TAMPERE UNIVERSITY OF TECHNOLOGY

MIKKO VIINIKAINEN

COMPUTER-AIDED BILATERAL TELEOPERATION OF MANIPULATORS

Master of Science Thesis

Examiner: Professor Jouni Mattila
Examiner and topic approved in the
Automation, Mechanical and
Materials Engineering Faculty
Council on 7.11.2012

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master's Degree Programme in Automation Technology
VIINIKAINEN, MIKKO: Computer-Aided Bilateral Teleoperation of Manipulators
Master of Science Thesis, **73** pages, 12 appendix pages
January 2014
Major: Machine Automation
Examiner: Professor Jouni Mattila
Keywords: Shared Control, Bilateral Teleoperation, Virtual Fixtures, Haptics, DTP2

Haptic bilateral teleoperation is often a challenging and mentally demanding job for the operators of robot control systems. It is especially difficult in cases such as the remote maintenance of the ITER divertor region. The difficulty of the ITER divertor maintenance hails from a multitude of reasons: the residual radiation level of the ITER reactor during a shutdown is too high for any human access, the maintenance tunnels of the divertor are confined, the operators have to operate heavy loads in delicate tasks, and only a limited number of radiation tolerant cameras are available for providing video feedback. In addition, most of the maintenance work cannot be automated because of the dynamic nature and complexity of the tasks.

Haptic shared control systems can be used for reducing the amount of mental and physical workload perceived by the operators of remote maintenance systems. To reduce the workload, a haptic shared control system assists the operators by generating virtual forces based on the virtual models of the teleoperation environment and sensor data from the slave manipulator. The generated assistance forces are laid over the force feedback signals from the teleoperation environment. The assisting forces can e.g. guide the operators along optimal paths and prevent collisions in the teleoperation environment. In addition to the reduction of the operator workload, teleoperation tasks also become faster and safer with haptic shared control.

This thesis investigates the implementation techniques and theory of haptic bilateral teleoperation and shared control systems. Based on the theoretical analysis, an experimental haptic shared control system, called the Computer Assisted Teleoperation (CAT) was developed. The intention of CAT is to assist the remote handling (RH) system operators of the Divertor Test Platform 2 (DTP2) in ITER remote maintenance research.

The effectiveness of CAT is evaluated in a teleoperation experiment performed with a 6 DOF Water Hydraulic MANipulator (WHMAN) developed for the ITER divertor maintenance. The results of the experiment gives directive indication that the CAT system improves the execution times of a bilateral teleoperation task and simultaneously reduces the workload perceived by the operators of the system.

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO
Automaatiotekniikan koulutusohjelma
VIINIKAINEN, MIKKO: Computer-Aided Bilateral Teleoperation of Manipulators
Diplomityö, 73 sivua, 12 liitesivua
Tammikuu 2014
Pääaine: Koneautomaatio
Tarkastaja: Professori Jouni Mattila
Avainsanat: Haptiikka, Etäoperointi, DTP2

Kaikki rakenteilla olevan ITER-fuusioreaktorin huoltotyöt joudutaan tekemään etäoperoitujen robottien avulla reaktorirakenteiden korkean radioaktiivisen säteilyn vuoksi. Huoltotyöt ovat teknisesti erittäin haastavia, koska käytettävät huoltotunnelit ovat ahtaita ja pimeitä, roboteilla käsiteltävät taakat ovat hyvin raskaita ja vaaditut voimat suuria. Huoltotehtävien monimutkaisuuden ja dynaamisuuden vuoksi suurinta osaa huoltotoimenpiteistä ei voida automatisoida. Huoltorobottien ohjaajien työtä vaikeuttaa edellä mainittujen seikkojen lisäksi myös saatavilla olevan videokuvan heikko laatu, joka pakottaa ohjaajat turvautumaan robottien haptiseen takaisinkytkentään ja virtuaalimallien käyttöön.

Huoltorobottien ohjaajien vaativaa työtä voidaan helpottaa luomalla keinotekoisia, virtuaalimalleihin perustuvia, tuntoaistimuksia ohjaajille. Nämä keinotekoiset voimat luodaan ohjelmallisesti yhdistämällä etäoperointiympäristön virtuaalimallien ja huoltorobotin tarjoamaa anturi-informaatiota. Keinotekoinen voima-avuste lisätään robotin haptisen takaisinkytkennän päälle. Voima-avuste voi esimerkiksi opastaa ohjaajan optimaalisille liikeradoille ja vastustaa ohjaajan liikkeitä, jotka saattaisivat aiheuttaa törmäyksiä etäoperointiympäristön kanssa.

Työssä käsiteltyjä teorioita soveltaen kehitettiin virtuaalisia voima-avusteita tuottava järjestelmä nimeltä ”CAT”. Järjestelmällä pystytään luomaan etäoperointijärjestelmän käyttäjää ohjaavia sekä käyttäjän virheliikkeitä estäviä virtuaalisia voimaopasteita. Opasteiden avulla etäoperointitehtävistä voidaan tehdä huomattavasti helpompia, nopeampia ja turvallisempia.

Tässä diplomityössä kehitettyä CAT-järjestelmää on käytetty menestyksekkäästi ITER-diverttorin huoltotesteissä DTP2-ympäristössä. Työssä esitellään järjestelmän toteutuksen keskeisimmät tekniset ratkaisut. Lisäksi järjestelmän tehokkuutta arvioidaan testeissä, joissa testikäyttäjät suorittavat ITER-diverttorille suunniteltuja etäoperoitavia huoltotoimenpiteitä DTP2-testiympäristössä. Testin tuloksena saadaan suuntaa-antava arvio, jonka mukaan CAT-järjestelmä parantaa huoltotoimenpiteen suoritusaikoja ja pienentää käyttäjän kokemaa työkuormitusta.

PREFACE

This master of science thesis was carried out at the department of Intelligent Hydraulics and Automation at Tampere University of Technology. The research work described in this thesis is a part of the ITER divertor remote maintenance research effort that has been carried out by the department.

I would like to express my gratitude to my instructor Professor Jouni Mattila for the opportunity to work at IHA and for the possibility of writing this thesis. I'm also grateful to Pekka Alho for the guidance and advice he has given me throughout the thesis process. My thanks for all my colleagues at IHA and DTP2 for the inspiring work environment and friendship.

I also want to thank my parents and my brother for their invaluable support during my studies.

Finally, thank you Elina for all your love and support.

Tampere 21.1.2014

Mikko Viinikainen

CONTENTS

1. Introduction 1
2. Bilateral Teleoperation 4
2.1 Background of Teleoperation 4
2.2 Stability and Transparency 6
2.3 Impedance 6
2.4 Impedance and Admittance Manipulators 7
2.5 Bilateral Teleoperation System Architectures 7
2.5.1 Position-Position Architecture 9
2.5.2 Position-Force Architecture 11
2.5.3 Four-Channel Architecture 12
3. Haptics and Teleoperation 15
3.1 Haptic Rendering 16
3.1.1 Human Somatosensory System 16
3.1.2 Haptic Interfaces 17
3.1.3 Collision Detection 18
3.1.4 Virtual Force Generation 20
3.2 Virtual Walls 22
3.2.1 Implementation 23
3.2.2 Force Generation 24
3.3 Guiding Virtual Fixtures 26
3.3.1 Implementation 26
3.3.2 Force Generation 28
4. DTP2 Computer-Aided Bilateral Teleoperation Control System 29
4.1 Bilateral Teleoperation Control System 29
4.1.1 DTP2 Software Architecture 30
4.1.2 Slave Device 32
4.1.3 Master Device 33
4.1.4 Bilateral Teleoperation Implementation 34
4.2 CAT Design and Implementation 35
4.2.1 System Analysis 35
4.2.2 Object Analysis 38
4.2.3 Development Environment 40
4.2.4 Interfaces and Data Content 41
4.2.5 Virtual Paths 42
4.2.6 Collision Detection 44
4.2.7 Virtual Walls 45
5. DTP2 Computer-Aided Teleoperation Experiment 48

5.1 Teleoperation Test 48
5.2 System Configuration 50
5.3 Procedure 51
5.3.1 Task Execution Times and Accuracy 52
5.3.2 Operator Workload 53
6. Conclusions 57
References 59
A.Appendix 1: Top-Level Architecture of the ITER RHCS 62
B.Appendix 2: CAT DDS Interfaces 63
C.Appendix 3: DTP2 DDS Interface Definitions 64
D.Appendix 4: DTP2 RHCS QoS Settings 66
E.Appendix 5: TLX Rating Scale Definitions Sheet 68
F.Appendix 6: TLX Workload Comparison Cards and Rating Sheet 69
G.Appendix 7: TLX Subject Instructions 72

ABBREVATIONS AND NOTATION

4C	Four-Channel bilateral teleoperation architecture
AABB	Axis-Aligned Bounding Box, a simple bounding volume commonly used in collision detection applications
CAT	Computer Assisted Teleoperation, a prototype haptic shared control system developed for the DTP2 bilateral teleoperation control system
CLS	Cassette Locking System
CMM	Cassette Multifunctional Mover
DDS	Data Distribution Service, the OMG specification for a publish/subscribe middleware
DH	Denavit-Hartenberg (parameter)
Divertor	Divertor is a term used for the bottom part of a tokamak type fusion reactor. The main purpose of the divertor is to extract helium ash from fusion plasma and to dissipate the heat energy produced by the neutron flux resulting from fusion reaction
DOF	Degrees Of Freedom
DTP2	Divertor Test Platform 2
EC	Equipment Controller, a low level robot control software developed for the DTP2 teleoperation control system
F4E	Fusion For Energy
FOV	Field Of View
FRVF	Forbidden-Region Virtual Fixture
GUI	Graphical User Interface
GVF	Guiding Virtual Fixture
Haptics	The word is derived from the Greek word haptesthai, meaning related to the sense of touch. In the context of robotics, generation of tactile and kinesthetic sensings in order to simulate interaction between humans, robots and real, remote or simulated environments
HIP	Haptic Interface Point

HMI	Human Machine Interface
IDL	Interface Definition Language
IHA	Department of Intelligent Hydraulics and Automation
IHA3D	Virtual environment visualization software developed at IHA
ITER	International fusion power research project aiming to prove the viability of fusion as an energy source
LabVIEW	Laboratory Virtual Instrumentation Engineering Workbench, developement environment for the graphical programming language called G.
LAN	Local Area Network
LGPL	GNU Lesser General Public License
MIS	Minimally Invasive Surgery
NASA	National Aeronautics and Space Administration
OBB	Oriented Bounding Box, a simple bounding volume commonly used in the collision detection applications
ODE	Open Dynamics Engine, an open source physics engine
OS	Operating System
P-F	Position-Force bilateral teleoperation architecture
P-P	Position-Position bilateral teleoperation architecture
PD	Proportional-Derivative (controller)
RAM	Random Access Memory
RH	Remote Handling
RHCS	Remote Handling Control System
ROViR	Remote Operation and Virtual Reality, an international research centre that focuses on the development and commercialization of remote handling and virtual technology
TCP	Tool Center Point
TLX	NASA Task Load Index

TUT	Tampere University of Technology
VF	Virtual Fixture
VR	Virtual Reality
VTT	Technical Research Centre of Finland
WHMAN	Water Hydraulic MANipulator, a prototype 6-DOF manipulator, developed at IHA

1. INTRODUCTION

Teleoperation is a technology that allows people to work in environments that are far beyond the limitations of our physical bodies. Without teleoperation technology, tasks such as handling nuclear waste, or exploring the deep sea, would be extremely difficult and dangerous for us. Nevertheless, these kind of tasks are necessary so that we can ensure our own safety or satisfy our endless curiosity. Teleoperation systems are also in a vital role for the future of fusion energy production and the ITER fusion reactor, which is currently being built in the Southern France. ITER is a critical step towards the commercial production of fusion energy which, if successful, has a promise of putting a definitive end to the global warming, air pollution and fears of power source exhaustion. However, the path to this goal is long and paved with technical challenges. The remote maintenance of the fusion reactors, using teleoperated systems, is not the least difficult one of those.

Teleoperation can be a challenging and mentally demanding job for the operators of the remote handling devices. And it is especially challenging in an environment such as the ITER divertor[1] region. Due to material erosion, divertor cassettes have to be replaced several times during the expected lifetime of the ITER facility [22]. This has to be done completely with teleoperated devices through the maintenance ports of the reactor. All human access is forbidden to the reactor, because the residual radiation level of the fusion reactor during a shutdown is lethal.

The remote maintenance of the ITER divertor is particularly challenging because the maintenance tunnels of the divertor are confined, pitch black and the operators have to be able to operate heavy loads and implement delicate tasks. Also, only a limited number of radiation tolerant cameras can be used for video feedback. Deployment of these cameras for optimal field of view (FOV) is a tedious task because of the space restrictions in the teleoperation environment. In addition, most of the maintenance work cannot be automated because of the dynamic nature and complexity of the tasks.

Virtual models and techniques can be used to reduce the amount of mental and physical workload perceived by the operators of the remote maintenance systems and make teleoperation tasks faster and safer. Especially haptic shared control

[1]Divertor is the bottommost part of the ITER fusion reactor. It consists of 54 modules, called cassettes. Each of the cassettes weighs approximately 8-9 tons. The main purpose of the divertor is to extract excess heat, helium ash and other impurities from the reactor.

systems have been demonstrated to improve teleoperation results significantly (e.g. [1, 19, 25, 27, 33]). Therefore these systems can make a significant contribution for the success of the tokamak based fusion technology which is dependent on the efficient remote maintenance of reactors.

This thesis introduces general theories related to the bilateral teleoperation and haptic shared control systems. The thesis also describes the development process of the haptic shared control system, called Computer Assisted Teleoperation (CAT). The CAT system was developed for assisting the operator teams working at the Divertor Test Platform 2 (DTP2). The DTP2 is a test environment used for the experimental divertor region remote maintenance research of the ITER fusion reactor.

Implementation of a haptic shared control system is a combination of software engineering and control theory. These are also the main themes for this thesis. Having a human physically in the closed loop system provides a special challenge for both the control and the software design. The challenge mostly originates from the need to accurately imitate the nature with a robot or a haptic device and from the fluctuating dynamics of people.

Another point of focus of the thesis is in the bilateral teleoperation control system architectures and the technologies used at the DTP2. CAT is a part of the distributed bilateral teleoperation control system of the DTP2 and interacts with other parts of the system. The surrounding bilateral control system sets requirements for the haptic shared control system and vice versa. The DTP2 control system architecture has two different bilateral teleoperation implementations for different teleoperation situations. One of the implementations is a traditional force feedback control and another is an adapted four-channel architecture that is loosely based on the theory presented in [20].

This thesis was written as an extension to an ITER divertor maintenance related research project called F4E-GRT-143 - Divertor RH Design Updates and DTP2 Phase 2 Testing. The project was funded by F4E (Fusion for Energy), EURATOM-TEKES and TUT (Tampere University of Technology). The goal of the project was to implement, identify and test upgrades to the RH (Remote Handling) equipment and the control systems of the DTP2 facility. The research work was carried out as cooperation between the VTT (Technical Research Centre of Finland) and IHA-TUT (Department of Intelligent Hydraulics and Automation). During the project, several new subsystems were implemented to the prototype DTP2 control systems. One of the new subsystems was the CAT system that is the subject of this thesis.

This thesis consists of the following parts: chapters 2 and 3 present the central theory of bilateral teleoperation and shared control systems. Chapter 2 introduces the basics of the bilateral teleoperation system control theory and architectures. Chapter 3 introduces the theory of haptic and shared control systems.

Chapter 4 describes the software and control system design and the implementation that were done for the CAT system of the DTP2 project. The chapter also includes description about the integration of CAT to the distributed DTP2 control system.

Chapter 5 presents an experimental study that was conducted to evaluate the achieved increase of operator performance with DTP2 CAT in one of the divertor maintenance tasks. In the experiment 10 test operators repeated the maintenance task with and without the CAT system. The test was performed using a full size Water Hydraulic MANipulator, developed for the divertor maintenance, and the DTP2 bilateral teleoperation control system. Discussion of the results of the project and drawn conclusions are presented in chapter 6.

2. BILATERAL TELEOPERATION

Teleoperation is a scientific term for the remote control of technical devices. The definition covers all the systems where devices are controlled from some distance, but most commonly the word is used for mobile and robotic applications where the operator is far away from the remote manipulator or vehicle. When a teleoperation system also offers force feedback functionality to the operator it is called a bilateral teleoperation system. Current teleoperation systems are most commonly used for medical, space exploration, dangerous materials handling, mining or military applications. However, the field offers great possibilities for applications in many other areas of engineering in the future.

Modern teleoperation systems are complex and composed of several hardware and software modules modules offering varying functionality. In addition to the robot control functionality, modern teleoperation control systems usually produce multi-modal[1] feedback from the teleoperation environment. Other supporting systems can include, for example, task planning, virtual reality, augmented reality and artificial feedbacks. This chapter introduces the essential theory and the common control architectures related to teleoperation systems and especially the haptic bilateral teleoperation.

2.1 Background of Teleoperation

The origins of teleoperation are in the invention of the radio technology and Nikola Tesla, who developed the first teleoperated device (a radio-controlled boat). This invention was patented in 1898 [5]. However, the bilateral teleoperation research only really got up to speed with the nuclear research where the need for the remote handling of radioactive materials quickly came apparent, after harmfulness of the radiation to humans was realized. The first modern bilateral teleoperation systems were built in 1940s by a research group, led by Raymond Goertz, in the Argonne National Laboratory, in the United States [34]. With these bilateral teleoperation systems, radioactive materials could be handled safely. These first systems consisted of mechanical manipulators, which were controlled by an operator behind a lead glass. The control device (master) used by the operator was identical to the manipulator (slave) on the other side of the glass. Movements of the master

[1] The term multi-modal refers to the different human senses.

device were relayed to the slave manipulator with a mechanical linkage. Through the mechanical linkage the operator was also able to feel the forces acting on the slave manipulator. The first modern master-slave teleoperation system is illustrated in Figure 2.1.

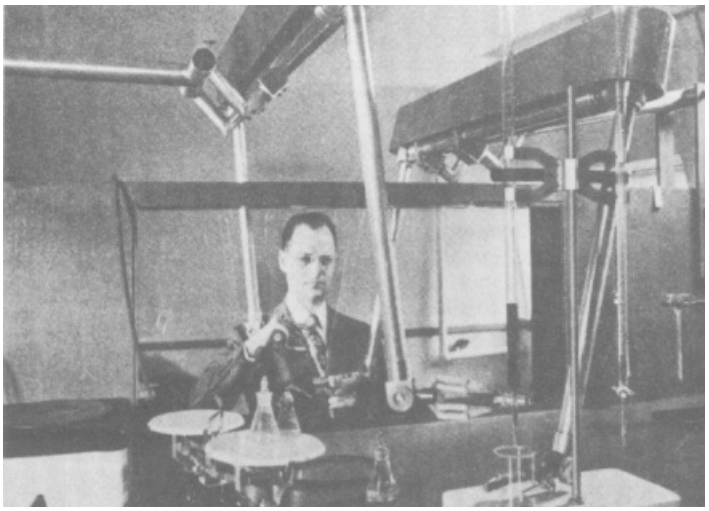


Figure 2.1: Raymond Goertz demonstrating the first master-slave manipulator [34].

The need for master-slave teleoperation systems and the basic concept has lasted over the decades but the mechanical devices and linkages have been replaced with electrical, pneumatic and hydraulic solutions. The control of the modern day bilateral teleoperation systems are, almost without an exception, implemented using computers and electronic communication links. Figure 2.2 illustrates the general idea of the modern teleoperation systems.

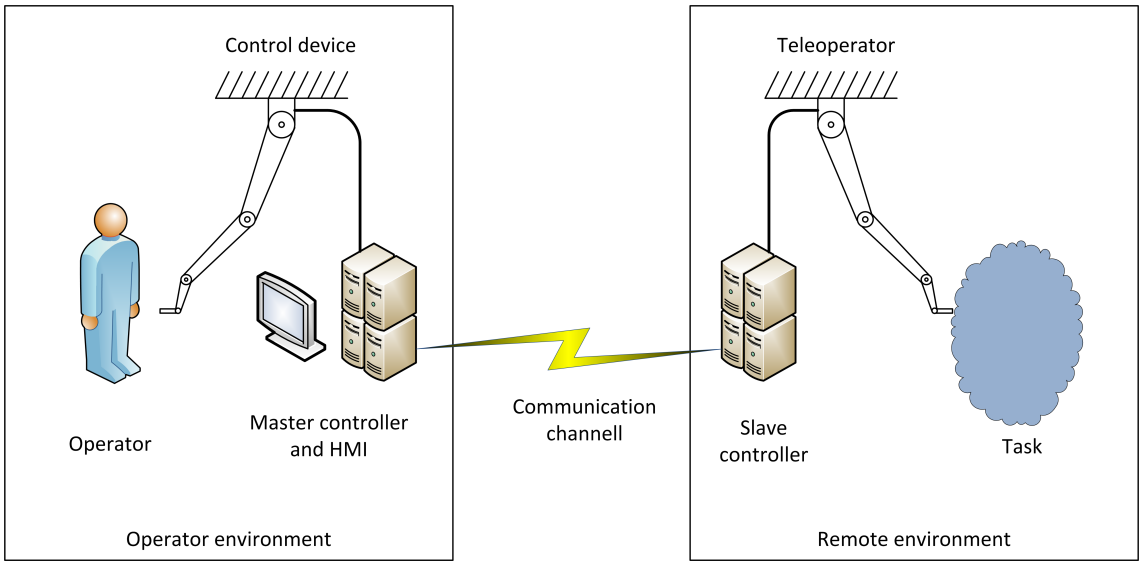


Figure 2.2: Concept of a modern teleoperation system.

Electrical actuation and software based control systems allows the teleoperation

distances to be vastly greater than the distances allowed by mechanical linkages. The main drawbacks of the electrical teleoperation systems are the cost, caused by the overall complexity of the systems, and the technical challenges caused by delays in the communication links.

2.2 Stability and Transparency

Transparency and stability are the two major challenges of the modern bilateral teleoperation systems. The term "transparency" means the degree of telepresence[2] associated to a teleoperation system. Therefore, in a system with a perfect transparency the operator of the system should feel as if he was manipulating the task directly, without the manipulators between him and the task. Perfect transparency is of course impossible to achieve, but a good degree of telepresence guarantees the feasibility of the required manipulation task [4]. The transparency and stability requirements of the bilateral teleoperation systems often become troublesome with the fact that transparency and stability requirements tend to have contradicting effects to the systems. Usually an improvement of transparency makes the system more unstable and increasing the stability impairs the level of transparency [20].

Generally a good level of transparency in a bilateral teleoperation system is pursued by making the slave manipulator to follow the motions and forces of the master faithfully and vice versa. Exceptions to the rule are the bilateral teleoperation systems that are intended for the tasks that cause fatigue to the operator or require superhuman accuracy. If the forces required for manipulation task are physically too demanding for the operator the forces can be scaled down from the slave to master. Respectively, the rate of motions can be scaled to achieve greater accuracy levels. For example, the tasks done with minimally invasive surgery (MIS) systems are typically heavily scaled. The scaling of forces or movements naturally deteriorates the level of transparency that the teleoperation system can provide and thus is not desirable unless necessary.

2.3 Impedance

The feel of different objects or materials can be measured using mechanical impedance (Z). From a physical point of view the mechanical impedance measures how much a structure resists motion when subject to a certain force. Therefore, when a telemanipulator comes to contact with its environment the robot feels the structure with an impedance (Z_e):

$$Z_e = \frac{F_e}{V_e}. \tag{2.1}$$

[2]Telepresence means that the operator receives information about the teleoperator and the task environment which allow the operator to feel as if he was physically present at the remote site.

Where F_e is the force applied to the structure and V_e is the speed of the slave robot.

In a teleoperation system that offers perfect transparency the operator would feel exactly the same impedance as the slave manipulator and therefore a system with perfect transparency would have to satisfy the condition:

$$Z_e = Z_t, \tag{2.2}$$

where Z_t is the impedance felt by the operator. In practice, dynamics of the operator and especially the environment vary drastically compromising both the stability and the transparency. Moreover, the communication delays further complicate the controller design problem [20]. Therefore a good balance between stability and transparency is required [16].

2.4 Impedance and Admittance Manipulators

Robot manipulators are divided into two categories: the admittance and the impedance devices. The category of a manipulator depends on whether the output magnitude of the device is force or velocity. An impedance device is controlled with a force input message that the device applies to its environment. The applied force results into a change in position of the manipulator. Respectively an admittance device is controlled with position or velocity commands that the robot tries to reach. While changing its position an admittance device exerts a certain force to the operating environment. This force is considered as an output of the manipulator. The choice between the admittance and impedance approaches for designing a manipulator is done early in the manipulator design process and has profound implications to the hardware and software design in the later phases.

The admittance type manipulators tend to be strong, accurate and fast, making them ideal industrial robots. The cost of these advantageous attributes is the low backdrivability of the manipulator. The lack of backdrivability is a result from high gear reductions of electric motors or incompressibility of hydraulic fluids. The impedance type robots on the other hand are easily backdrivable and adapt well to different environments making them the natural choices for master manipulators. [15]

2.5 Bilateral Teleoperation System Architectures

The goal of a typical bilateral teleoperation system is to reproduce the movements of the master manipulator with the slave manipulator and to simultaneously reflect the dynamics of the teleoperation environment to the master device. There are several different control system architectural approaches how this condition is generally pursued. Following paragraphs introduce some of the common architectures.

Notation of the bilateral teleoperation architectures in this thesis follows the architecture notation style introduced by D.A. Lawrence in [20]. In the Lawrence's general 4-channel teleoperation architecture both the master and the slave manipulators have their own force or position/velocity controllers. Quantity that is being controlled depends on the manipulator and whether it is an admittance or impedance device. In addition outer control loops are added using communication channels of the system. Stability of this kind of control system can be analysed using the network theory methods. The control aspects of the 4-channel teleoperation architecture are introduced in more detail in subsection 2.5.3. Figure 2.3 visualizes the general bilateral teleoperation architecture. The figure also illustrates forces reflected by both environments to the system.

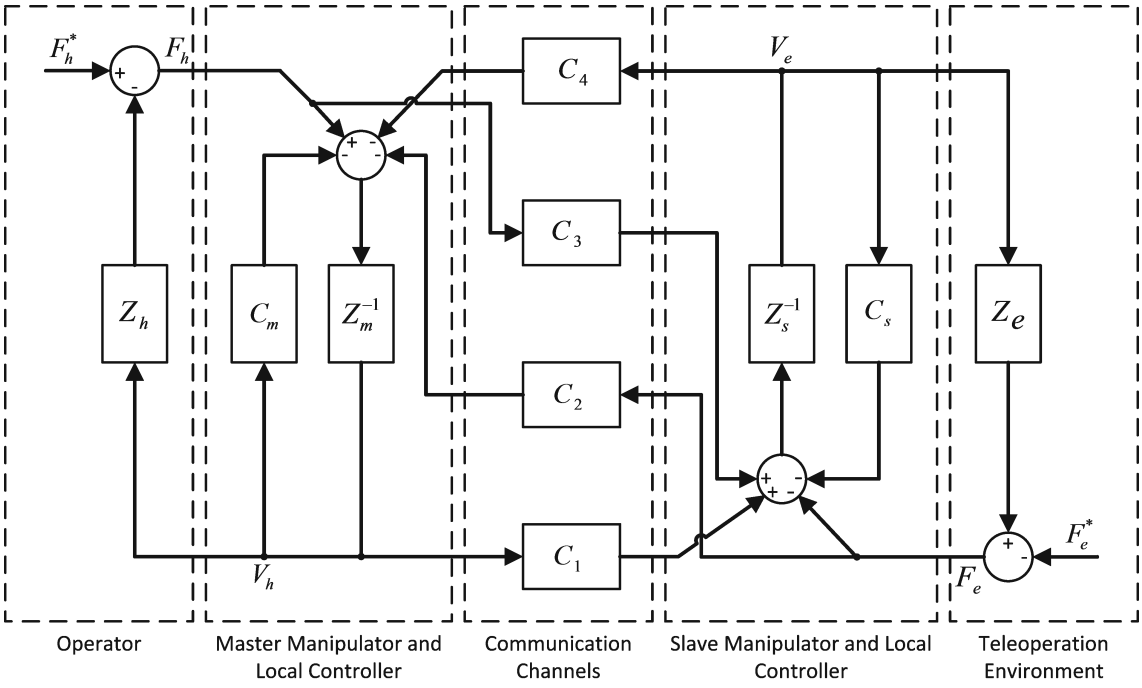


Figure 2.3: General bilateral teleoperation control system architecture by D.A. Lawrence [20]. The architecture includes force and velocity channels to both directions.

Symbols in the picture are:

- Z_h, operator impedance
- Z_e, environment impedance
- F_h^*, operator exogenous force input
- F_e^*, environment exogenous force input
- C_m, master local position controller
- Z_m, master impedance

- Z_s, slave impedance
- C_s, slave local position controller
- C_1, master coordinating force feedforward controller
- C_2, slave force feedforward controller
- C_3, master force feedforward controller
- C_4, slave coordinating force feedforward controller.
- V_h, master manipulator velocity.
- V_e, slave manipulator velocity.

The general architecture has several variations that have been applied successfully to the real-world bilateral teleoperation systems. Most common ones are the position-position and position-force architectures. The position-position architecture is sometimes also called the coordinating force architecture and the position-force architecture is commonly called force feedback. If the position of the master device is interpreted as a velocity command for the slave, the method is called rate control [28]. The architecture names denote the communication channels used in each case.

Bilateral teleoperation systems that contain haptic shared control functionalities (which are described in detail in chapter 3) have artificial force signals combined to the force feedback signals. Artificial forces can be added in several places of the architecture but propably the easiest way is to add the artificial signal to the real force measurements (F_e or F_h) depending on wether the assistance is added to the master or slave side of the system. In this case the artificial force signal appears as interference similar to the contact force or user applied force.

2.5.1 Position-Position Architecture

The position-position (P-P) bilateral teleoperation architecture is the simplest case when it comes to the bilateral teleoperation architectures. It is usually the most cost effective solution to implement as well, because the only hardware requirement of the architecture is the position sensing on both manipulators. Most electrical and hydraulic manipulators are equipped with the position sensing out of the box. Another benefit of this architecture is that it can be shown to be passive [35]. The passivity (in engineering contexts) means that a component can consume energy but can not produce or increase it. In most cases passivity can be used to demonstrate that a passive circuit will be stable under specific criteria. This quality is particularly useful when studying stability of complex systems.

The contact force of a manipulator is usually proportional to the difference between the desired and actual machine positions [35]. The P-P architecture takes advantage of this property of manipulators by instructing both, the master and slave manipulators to track the positions of each other. Therefore, when the teleoperation system is not able to match the positions, the resulting difference between the manipulator positions is perceived as a force that drives the positions of manipulators to the same value. Figure 2.4 depicts the whole P-P control scheme.

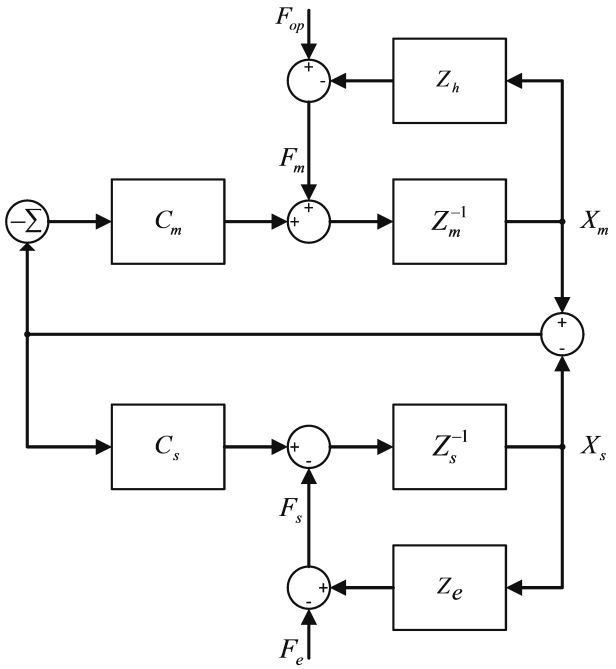


Figure 2.4: Block diagram presentation of the position-position bilateral teleoperation architecture.

Blocks of the diagram denote different components affecting the control system. C_m and C_s are the position controllers for both the master and the slave. Z_m is the master manipulator and Z_h is the impedance of the operators hand. Z_s is the impedance of the slave manipulator and Z_e is the working environment of the manipulator. In a static situation the components of the diagram can be defined in a transfer function form as follows:

$$Z_m = M_m s, \tag{2.3}$$

$$C_m = B_m + \frac{K_m}{s}, \tag{2.4}$$

$$Z_s = M_s s, \tag{2.5}$$

$$C_s = B_s + \frac{K_s}{s}, \tag{2.6}$$

$$Z_h = M_h s + B_h + \frac{K_h}{s}, \tag{2.7}$$

$$Z_e = M_e s + B_e + \frac{K_e}{s}. \tag{2.8}$$

Where M_m and M_s are the masses of the master and slave manipulators. M_h, B_h and K_h are the mass, damper and spring coefficients of the hand of the operator. Respectively M_e, B_e and K_e are the mass, damper and spring coefficients of the teleoperation environment. Controllers of the P-P architecture are usually PD-position controllers, that act similar to a spring and damper ($K_{m,s}$ and $B_{m,s}$) in natural phenomena.

The most significant issue with the usage of the P-P architecture is that the operator feels extra inertia when using the system in free space. This makes the teleoperation system feel sluggish. Also in the extremes of the relayed impedance the operator feels the dynamics of the teleoperation system and not the task [20].

2.5.2 Position-Force Architecture

The position-force (P-F) architecture (traditional force-feedback) is the most intuitive one of the teleoperation architectures. The principle of the architecture is that the slave manipulator accurately follows movements of the master manipulator, and the master manipulator accurately repeats the forces sensed by the slave manipulator. In this case, the slave manipulator has to be equipped with a force sensor that senses the forces and torques that are reflected from the teleoperation environment. Implementing a teleoperation system with this architecture is generally more expensive than the position-position case because force sensors are rather expensive and usually have to be installed specially for the bilateral teleoperation needs. Figure 2.5 illustrates the concept of the position-force architecture with a block diagram.

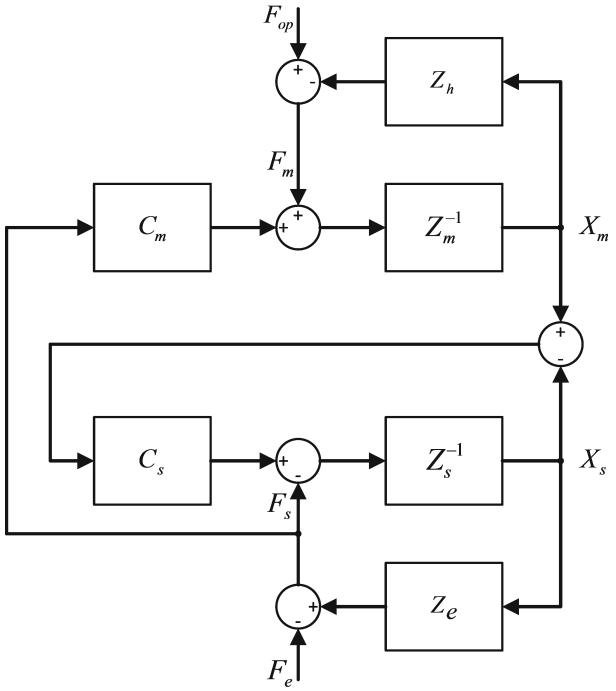


Figure 2.5: Block diagram presentation of the position-force teleoperation architecture, including external forces.

From the control system point of view, the P-F architecture is fairly similar to the P-P architecture. The difference is that the controller of the master manipulator is a force controller rather than position controller and the set point is changed accordingly. The force controller is also usually just a scalar gain instead of the PD-controller of the P-P architecture. Blocks of the block diagram are defined similar to the position-position architecture with the exception of:

$$C_m = \frac{K_m}{s}. \tag{2.9}$$

A typical problem of the bilateral teleoperation systems implemented with the P-F architecture is instability. Presence of a substantial time delay in the communication links is well known to make these bilateral teleoperation systems unstable, unless the feedback force gain (M_m) is dampened significantly. The additional dampening in the feedback alters the feeling that the operator senses through the teleoperation system, effectively reducing the transparency of the system [20].

2.5.3 Four-Channel Architecture

Both of the aforementioned teleoperation architectures can produce rather good teleoperation results in terms of the transparency. However, the success of these architectures is largely dependent on the application and used hardware. Better results with the transparency-stability trade-off can always be achieved with the

4-channel (4C) architecture. This architecture utilizes position and force/torque channels in both directions, which improves the transparency of the system [20]. The main drawback of the 4C implementations is the price. Force/torque sensors are required in both manipulators and the overall complexity of the system makes it more demanding to develop and tune.

In theory, the 4C architecture is capable of delivering perfect transparency for the teleoperation system with unlimited transmitted impedance. However, the limitations of the physical world render the perfect transparency impossible even for the 4C architecture. Figure 2.6 illustrates the 4C architecture in a block diagram form.

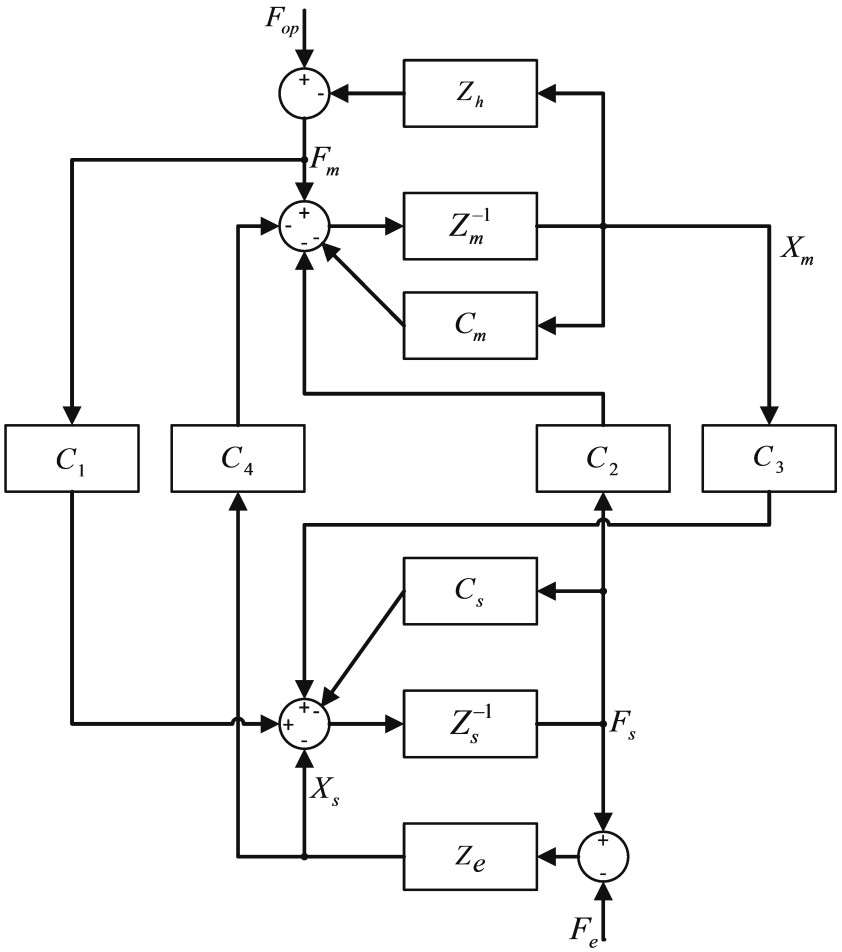


Figure 2.6: Block diagram representation of the 4-channel architecture including external forces. In this representation the master is an impedance device and the slave is an admittance device.

The architecture represented in the figure has a slight modification to the original Lawrence's architecture presented in section 2.5. The architecture above has an impedance type master device and an admittance type slave device, which is usually the case when large slave devices are operated. In the 4C architecture the local position

controller of the master (C_m) is a PD-controller similar to the position-position architecture. Position controller of the slave is implemented with an impedance filter. The architecture also includes separate controllers for each of the communication channels. Communication channel controllers are called: the master coordinating position feedforward controller (C_1), slave force feedforward controller (C_2), master position feedforward controller (C_3) and slave coordinating force feedforward controller (C_4). Both coordinating feedforward controllers (C_1 and C_4) are impedance filters and the force feedforward controllers (C_2 and C_3) are scalar gains. According to [20] perfect transparency could be achieved by tuning the communication channel controllers as follows:

$$C_1 = Z_s + C_s, \tag{2.10}$$

$$C_2 = 1, \tag{2.11}$$

$$C_3 = 1, \tag{2.12}$$

$$C_4 = -(Z_m + C_m). \tag{2.13}$$

Another common issue with the teleoperation systems is the effect of time delay in the communication channels. Various methods for eliminating the problems caused by the time delay have been developed. Most of the experimentally successful approaches are based on the scattering (wave variable) transformation techniques [17]. In the context of this study, the delay in communication channels was not a problem and therefore the research was restricted to the basic case of the 4C teleoperation architecture along with the P-F-architecture.

3. HAPTICS AND TELEOPERATION

The word haptics originates from the Greek word haptesthai, which means related to the sense of touch [15]. In psychological and physiological contexts haptics refers to the study of the human sense of touch, whereas in technical contexts haptic technology is used for creating sensations for the human operators operating with mechanical devices. The haptic sensations can be generated with software, on the basis of real, remote or virtual environments.

In shared control teleoperation a computer tries to assist the operator in accomplishing teleoperation tasks. In the haptic shared control systems the assistance is implemented as a software system that offers haptic assistance to the bilateral teleoperation operators. The assistance consists of software-generated force and position signals that are applied to the control devices used by the operators. These signals can, for example, prevent the operators from entering certain subspaces in the teleoperation environments, or guide the operators to certain locations. Haptic shared control systems have been previously demonstrated to improve teleoperation results significantly (e.g. [27, 25, 33]).

The implementations of shared control systems usually rely on an abstract concept called haptic virtual fixtures (VF). This concept was first introduced by Rosenberg in [27]. Rosenberg defined virtual fixtures as an overlay of abstract sensory information on top of sensory feedback from the remote environment. The definition, proposed by Rosenberg, was not only limited to the software generated aids affecting the sense of touch. The definition also covered much larger array of means of assistance, such as audible aids. In this thesis however, only the haptic virtual fixtures are introduced.

As a metaphor of the benefit gained from the usage of haptic virtual fixtures, those are often compared to the real-world ruler: Making precise movements freehand, such as drawing a straight line, is difficult and imprecise even without a teleoperation system between the human and the paper. However, if a simple ruler is used for the task, it becomes much easier mentally and also much faster and more precise. Usage of the haptic virtual fixtures has similar effects in the the bilateral teleoperation context but possibilities for assisting the operator with computers are much greater.

This chapter introduces the theory related to the development of the haptic shared control system called CAT. The chapter introduces the general architecture and

techniques for haptic software applications. Also the sensory system of a human, how it senses touch and what requirements does this set for the interfacing technical system are briefly introduced. Last, the specific theoretical background of the haptic shared control systems is introduced.

3.1 Haptic Rendering

Haptic shared control implementations are in essence normal haptic software applications with tighter real-time requirements, safety considerations and interaction with the slave manipulator. Figure 3.1 depicts a basic architecture for a haptic application.

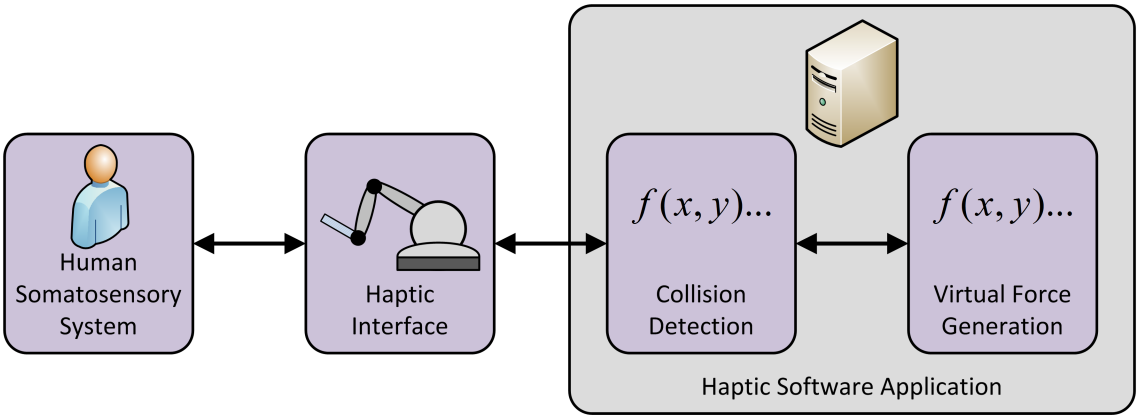


Figure 3.1: Basic architecture for a virtual reality application generating haptic feedback.

The rendering[1] of haptic sensations is a rather unique type of a human-machine interface (HMI). Whereas the typical visual and audible interfaces are unidirectional information flows (from simulation environment to the user), a haptic interface is bidirectional. The basic architecture presented in Figure 3.1 is broken down further in the following subsections to create an overall view of the systems used for implementing haptic shared control system software.

3.1.1 Human Somatosensory System

Haptic feedback affects the somatosensory system of a human. The somatosensory system combines several different methods of the human nervous system to create sensations. These are the sense of touch (tactile sense), body position (proprioception), movement (kinaesthetic sense), temperature and pain. The Kinaesthetic and proprioception senses are based on the ability to sense forces and displacements inside the muscles and tendons while the sense of touch means the ability to feel

[1] Rendering refers to the process by which artificially generated sensory stimuli are imposed on the user.

deformations of the skin. Haptic shared control systems and haptic applications only attempt to affect the proprioception and kinaesthetic senses but other areas of the somatosensory system are also stimulated.

The somatosensory system of a human is very advanced and especially sensitive in the hands. Tactile receptors of a hand are known to be able to sense frequencies up to 10 kHz [30] and displacements in the micrometer scale [15]. Because the hands are such a well-tuned mechanism, fooling the nervous system into believing that the artificial stimuli are real is a challenging task. If the update rate of the artificial force is too low, the operators can feel the discontinuities in the force signal.

The update rate of a haptic application also limits the achievable stiffness of the projected virtual surfaces, effectively dictating e.g. how hard a rigid wall really feels like. Therefore, a sufficiently high update rate of force generation is imperative to the haptic applications. However, a high update rate of forces means that there is less time for calculating the feedback signal, reducing the achievable detail of the feedback. For these reasons, haptic shared control systems have to compromise between the stiffness and the detail of the virtual fixtures. Fortunately the limitations are only generated by the available computational power and the effectiveness of algorithms. The performance limitations can therefore be circumvented by adding more powerful hardware and/or more efficient software to the system.

There are no firm rules for the required update rate of a realistic haptic application but 1 kHz is a very common choice. The 1 kHz update rate seems to be a fairly good compromise for permitting the presentation of reasonably complex objects with a reasonable stiffness [29].

3.1.2 Haptic Interfaces

Haptic interfaces come in various sizes and levels of sophistication. Some of the simpler designs are seen in the games console controllers that can produce vibrating kinaesthetic feedback, usually in one or two degrees of freedom. More complex haptic interfaces range from the table top commercial haptic devices to the exoskeleton mechanisms or body-based haptic interfaces, which a person wears on the arm or leg. The exoskeleton or body-based interfaces are typically heavy, clumsy and extremely expensive, which is why these kind of devices are rare. The haptic interfaces used in the bilateral teleoperation applications are commonly somewhere between the extremes in terms of complexity.

Robot applications usually utilize either custom made haptic devices or commercially available table mounted haptic devices. Most of the commercially available haptic devices are manufactured by either the Immersion corporation or Geomagic (formerly SenSable). Figure 3.2 is a photograph of the Phantom Omni haptic device which is a very popular low-cost six DOF haptic device manufactured by the

Geomagic.

Figure 3.2: Phantom Omni haptic device. The device can measure position of the haptic device handle in six degrees of freedom and produce feedback forces in three degrees of freedom. [12]

Similar to the robot manipulators, haptic devices can be divided to the admittance and impedance device categories. Haptic devices are one kind of robot manipulators after all. As mentioned in the section 2.4, the impedance type manipulators are well suited as master manipulators due to the low internal impedance and backdrivability. The impedance type devices are also much more simple to design and affordable to produce, than admittance devices, making them the most common haptic device type [29]. The drawbacks of the impedance type haptic devices are usually a small workspace and a low force output capability. Especially in the cases where the slave is large and powerful, the limited force and workspace of an impedance type haptic device is problematic for the operator telepresence.

3.1.3 Collision Detection

An efficient and reliable collision detection is of paramount importance for haptic assistance systems. The collision detection determines when a haptically controlled object touches another object in the virtual space and in which direction the collision affects to. This easily seems like a trivial task to implement but in reality is a rather difficult one.

The collision detection task can be divided into three parts: determining if, when, and where two objects come into contact. These three tasks increase in difficulty in roughly this order. Another factor to be considered, especially with the haptic assistance systems, is the requirement for a high force refresh rate and the real-time constraints that also cover the collision detection algorithms. For an application

such as a haptic assistance system, the collision detection may easily take most of the available computational power. Typically compromises between the collision detection detail and the update rate have to be done. [8]

Virtual models are typically constructed from polygons[2]. In computer graphics multitude of polygons are combined to polygon meshes that are well suited for rendering on a screen. Figure 3.3 presents an example illustrating the use of polygons to form a virtual model. Although polygons suit well the computer graphics, detecting a collision between polygon meshes is a very heavy task for a computer. To ease the computational load of calculating collisions, collision detection algorithms use bounding volumes in conjunction with the virtual models. Bounding volumes are simple geometric forms placed around the polygon meshes. The idea of the bounding volume usage is that detecting the collisions between simple objects, such as the balls or boxes, is computationally a much simpler task than the collisions between the polygon meshes. When the array of possible colliding shapes is limited, the collision detection algorithms can also be made much faster and efficient than the generic solutions for the problem. A collision detection algorithm using simple bounding volumes is accurate enough for most applications. [8]

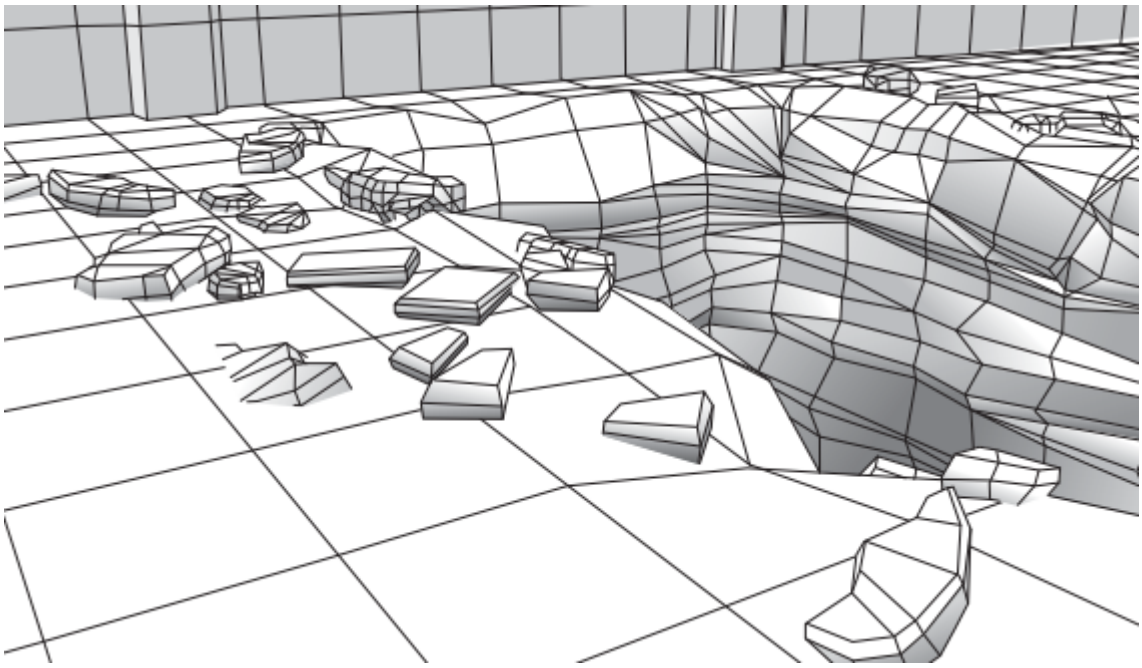


Figure 3.3: Polygon meshes used for constructing a 3D-model of a geographical formation[8].

There are several standard types of bounding volumes that vary in terms of required computational power and the offered detail of the collision detection. The

[2]Polygon is a 2D-shape that consists of straight lines and form a closed circuit. Multitude of polygons can be attached to each other from their edges to form 3D-surfaces.

DTP2 CAT-project uses oriented bounding boxes (OBB) for its collision detection. An OBB is a rectangular box related to an object with an arbitrary orientation. The OBB is a special case of the axis-aligned bounding box (AABB) which is otherwise similar rectangular box but the orientation of the box is fixed to the axis of the master object. Figure 3.4 is a visual example of the axis-aligned and oriented bounding boxes. Collisions between the AABBs are much lighter to calculate but the detail of the collision detection is rather poor. The OBB introduces a significant improvement to the quality of the collision detection result. Other well-known bounding volume types are: sphere, eight-direction discrete orientation polytope and convex hull [8].

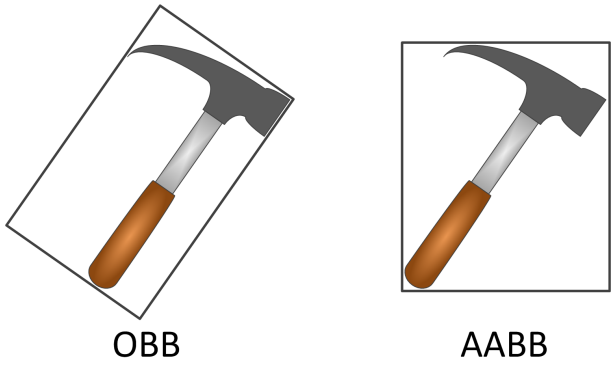


Figure 3.4: Two types of bounding volumes: an OBB (Oriented Bounding Box) and an AABB (Axis Aligned Bounding box). From the collision detection performance point of view an AABB is lighter to calculate but produces worse collision detection results.

Common to all the popular bounding volume shapes is the relative inexpensiveness of the collision testing computation and small memory usage. Advantageous properties for bounding volumes are also the simplicity of rotation and transformation functions. [8]

Several physics engines that include collision detection algorithms have been developed over the years. Therefore it is usually not necessary to develop custom made collision detection engines for applications. Some of the more famous physics engines are: Box2D, Bullet and Chipmunk. In this project a physics engine called the Open Dynamics Engine (ODE) was used. ODE is a community developed physics engine that is distributed under the LGPL license (GNU Lesser General Public License). ODE is designed for real-time collision detection and is highly stable, which makes it a suitable choice for the collision needs in a haptics assistance related project. ODE uses a C/C++ interface and also supports a wide variety of hardware platforms.

3.1.4 Virtual Force Generation

Work flow of a haptic application is depicted in the Figure 3.5. This figure also illustrates the relative positions of the kinematic functions and the hardware controllers

of the haptic device. The safety functions and other essential control functions are assumed to be integrated in to the controller block. A similar internal work flow also holds truth for the slave manipulators of teleoperation systems. Only the haptic rendering is replaced with a real world environment.

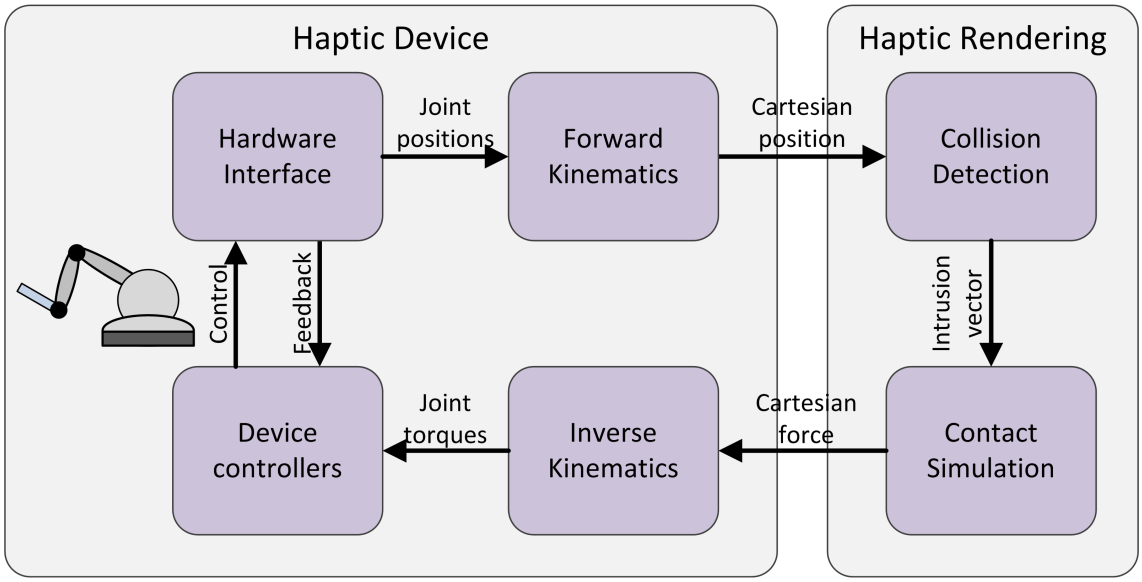


Figure 3.5: Flowchart model of a typical impedance-type haptic device feedback force rendering cycle.

In a haptic application, the position data of the haptic device joints is processed by the kinematics to produce the position of the device in the cartesian space coordinates. The collision detection is performed on basis of the acquired cartesian position of the control device and the virtual world. If the collision detection concludes that a collision occurred in the virtual world appropriate force calculation algorithms are triggered. These algorithms generate force and torque signals based on the rules defined by the user and the developer. The basis of calculating the contact forces F for the virtual collisions is usually the Hooke's law (spring system):

$$F = -Kx, \tag{3.1}$$

where x is the penetration vector and K is the spring constant. When the K term is set high enough the object in the virtual collision is starting to feel like a wall. The achievable stiffness of a wall is dictated by the dynamics of the haptic device and the update rates of the controllers. Especially in teleoperation systems, trying to achieve too high stiffness values tend to make the teleoperation systems unstable. In order to improve the stability in hard contacts with the haptic systems, damping is often added:

$$F = Kx + B\dot{x}, \tag{3.2}$$

where B is the damping coefficient. Usually the K and B are empirically tuned to generate a stable and high-performance operation [15].

Because haptic devices are usually constructed from joints connected with mechanical links the force information calculated in the cartesian space coordinates has to be transformed to the joint space which is used by the haptic devices. Usually all the actuators of a haptic device are revolutionary. In this case the desired torque commands for these actuators can be calculated with:

$$\tau = J^T f, \tag{3.3}$$

where τ is the torque of the actuators, J^T is the transpose of the haptic device Jacobian matrix and f is the desired force in the Cartesian space.

A common problem in the haptic rendering and teleoperation control systems is that most manipulators are equipped with sensors for measuring angles or displacements. However some control systems require the knowledge of manipulator speed or even acceleration. This means that the position has to be differentiated with the computer and doing that notoriously produces substandard signals. The quality of the velocity measurement is dependent on the sampling rate and can be compensated with the controller design or by using multiple sample differentiators. Using multiple samples for the differentiation introduces an additional delay to the control system which is undesirable for all haptic systems.

3.2 Virtual Walls

Haptic virtual fixtures can be divided into two categories, to the fixtures that attract the operator movements and to those that resist the operator movements. This section presents the theory behind the resisting virtual fixtures. For the sake of clarity, the term virtual wall is used exclusively within this thesis for describing the resisting virtual fixtures. Several other terms have also been used to represent the virtual walls in literature. These include, for example, forbidden-region virtual fixtures (FRVF), reactive virtual fixtures, virtual barriers and resisting virtual fixtures.

The most common teleoperation usage for a virtual wall is to forbid access to some areas of the workspace by virtually creating a protective barrier. However, few other usages for virtual walls have been introduced in the literature as well. For example in [27] Rosenberg used virtual walls for guiding the operator in a peg-in-hole task. Another possibility is to create a bidirectional virtual fixture that can be penetrated when a certain threshold is passed. After the penetretation threshold of a bidirectional virtual fixture is passed the fixture reversibly tries to keep the end-effector inside the fixture. This feature can be used e.g. for limiting the teleoperation workspace. A bidirectional virtual fixture has been introduced, in [26].

3.2.1 Implementation

Virtual walls can also be divided into two categories depending on the method used in the implementation. These are the impedance and admittance virtual fixtures. An impedance virtual fixture is the kind of virtual fixture that is usually described as a virtual wall. The virtual fixture generates resisting forces proportional to the amount of penetration in to the fixture. The push-back force signal is generated regardless of the user interaction when the virtual wall is penetrated. Naturally no resisting force whatsoever is generated if the manipulator does not penetrate the fixture. More detailed description of the force generation is provided in 3.2.2.

The most significant drawback of the impedance type virtual fixtures is that they are not passive instances. An impedance virtual wall produces energy internally, making it an active system which cannot guarantee stability via passivity. An impedance virtual wall can also cause distraction and possible safety issues if the operator changes his grip from the control device while the wall is penetrated. Figure 3.6 presents the concept of an impedance-type virtual wall where a surface generates resisting force when the manipulator enters in to the virtual wall.

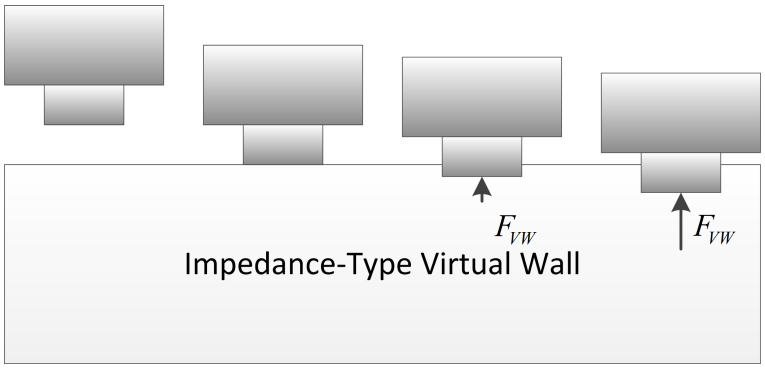


Figure 3.6: An impedance-type virtual wall. The push back feedback force F_{VW} is generated only if the virtual wall is penetrated.

Admittance-type virtual walls use a software generated proxy position for the force generation instead of the slave device position. Admittance virtual walls are sometimes also called the proxy-based virtual walls. The admittance wall prevents all penetration of the slave device in to the virtual fixture. In the admittance case the position of the slave manipulator always follows the position of the proxy. When the manipulator is moved in free space the proxy follows the master position and the slave coincides with the proxy. However, when the master penetrates a virtual wall the proxy will remain on the surface of the virtual fixture together with the slave. The control system tries to minimize the distance between the proxy and master by attracting the master towards the proxy position. How the attraction feels to the operator, can be tuned by changing the dynamics of the proxy. The proxy position

can not enter the virtual fixture and therefore the resisting force exists until the master is moved outside the virtual wall.

An important consideration when implementing a virtual wall application with the admittance architecture is the situation where the master position moves inside the virtual fixture and due to an edge the proxy suddenly has shorter distance to the master from another side of the virtual wall. In these cases the jumping of the proxy from one side of the wall to another has to be prevented. Figure 3.7 illustrates the usage of the proxy with an admittance virtual wall. Also the proxy jumping problem case is presented. In the figure the red dot illustrates the proxy position and the yellow dot is the end effector of the master manipulator.

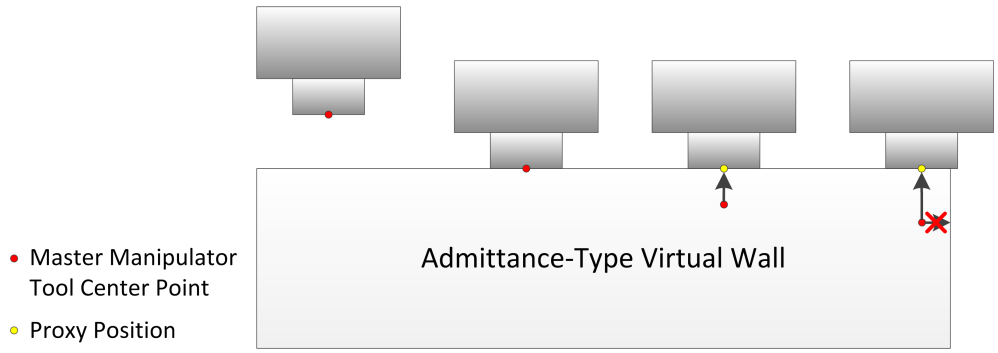


Figure 3.7: Illustration of an admittance type (proxy-based) virtual wall. The measured position of the master is marked with red dot and the position of the proxy with yellow dot. The right most case illustrates the possibility of proxy jumping near the edge of the virtual fixture.

It is possible to implement virtual walls to either the slave or the master side of the manipulator control system in order to achieve its purpose. Virtual walls can even be used on both sides simultaneously. Abbot [1] concluded that the slave-side virtual walls are more effective for rejecting disturbances on the slave side while maintaining the sense of telepresence for the user. And the master-side virtual walls are more effective for rejecting unintentional user commands into the forbidden region, while maintaining a sense of telepresence. The admittance virtual walls take away part of the operator control over the slave device which is contradicting with the general bilateral teleoperation goal of giving the operator best possible freedom of controlling the slave. Therefore the choice of virtual wall type is task dependent.

3.2.2 Force Generation

The equations in this paragraph are presented in one degree of freedom for clarity. The same equations can be applied in multiple degrees of freedom as well. Typically the impedance type virtual walls are defined with the spring model (Hooke's law),

along these lines:

$$F_t = \begin{cases} 0, & x_m < x_o \\ K_{VF} x_{i,} & x_m \geq x_o, \end{cases} \tag{3.4}$$

where F_t is the virtual wall force, x_m is the position of the manipulator, x_o is the position of virtual wall edge and x_i is the intrusion vector in to the virtual wall.

The Hooke's law is passive guaranteeing the stability of the impedance type virtual wall with continuous time controllers. However, a virtual wall implemented only with the Hooke's law and discrete controllers tends to become unstable at the higher spring stiffness values. The stability problems occur as a force jitter near the contact point of the virtual wall. The oscillation intensifies at the higher spring constant (K_{VF}) values, which are necessary in order to achieve stiffer virtual walls. The oscillations can also easily damage the motors of the impedance type haptic devices. Reason for the instability with the discrete controllers is the sampling rate of the computer which makes the virtual wall to turn on and off at slightly different locations [6]. The slave side virtual walls become unstable at lower stiffness values than the master side fixtures because the human hand adds damping to the system. However, the frequency of the jitter is higher than the human hand can produce consciously or unconsciously. Therefore the human hand can not remove the stability problem completely without additional help. To counter the instability caused by the discrete controller energy leaks several methods have been developed over the years. One of the simplest solutions is to add damping (B_{VF}) alongside the K_{VF} term of the Hooke's law.

$$F = \begin{cases} 0 & x_m < x_o \\ K_{VF} x_i + B_{VF} \dot{x}_{i.} & x_m \geq x_{o.} \end{cases} \tag{3.5}$$

The main difference between the admittance and impedance virtual walls is that an admittance type virtual wall does not allow the slave to have any movement in to the fixture. Therefore, the virtual force is implemented using a software generated proxy which usually coincides with the master position but does not follow the master in to the fixture. The control law in this case becomes following:

$$x_p = \begin{cases} x_m, & x_m < x_o \\ x_{o,} & x_m \geq x_{o,} \end{cases} \tag{3.6}$$

$$F_t = K_{tp}(x_p - x_m) - K_{tv}\dot{x}_m, \tag{3.7}$$

where x_p is the position of the proxy and K_{tp}/K_{tv} define the dynamics of the proxy.

Third method for implementing virtual walls is to scale down the movements of

the master with some constant α when the master enters the virtual fixture. This virtual wall type is also implemented using the proxy.

$$x_p = \begin{cases} x_m, & x_m < x_o \\ \alpha x_m, & x_m \geq x_o, \end{cases} \tag{3.8}$$

$$F_t = K_{tp}(x_p - x_t) - K_{tv}\dot{x}_t. \tag{3.9}$$

Abbot [1] concluded that none of the implementation techniques performed significantly better than others in his experiments with a fairly large sample quantity. He also suggests that the choice of the technology should rather be made on the basis of the task. For safety reasons the impedance type virtual walls should always be implemented only on the master side. For DTP2 CAT an impedance based approach of the master side virtual fixture was implemented. All the movements of the slave are scaled down in the DTP2 manipulator control system. The scaling factor remains the same on both the free space and the constrained motions. Also the control system force instability when in contact with the stiff virtual walls was reduced using damping:

$$F = \begin{cases} 0 & x_m < x_o \\ K_{VF}x_i + B_{VF}\dot{x}_i, & x_m \geq x_{o.} \end{cases} \tag{3.10}$$

$$\alpha x_m = x_s. \tag{3.11}$$

3.3 Guiding Virtual Fixtures

Guiding virtual fixtures (GVF) are the opposite to the virtual walls that were introduced in the section 3.2. The GVFs are geometric objects such as tubes, cones, cylinders or spheres, which are guiding the operator to specific points of interest or along a certain path in the teleoperation environment. Different kinds of guiding virtual fixtures can also be connected to form a more complex systems as in [19].

The most common application of the GVFs is a path that guides the operator to a specific point or generally through a path surrounded by objects where the manipulator should not collide. The path can be used as a safety precaution or for guiding the operators through optimal paths to increase the efficiency of teleoperation.

3.3.1 Implementation

GVFs can be either impedance or admittance type, similar to the virtual walls. The impedance type GVFs are potential fields that are always guiding the operator to a certain position or direction until the destination is reached or the fixture is turned

off by some other logic or rule. Figure 3.8 illustrates a two dimensional example of an impedance type GVF where the force generation is calculated using the Hooke's law (3.1).

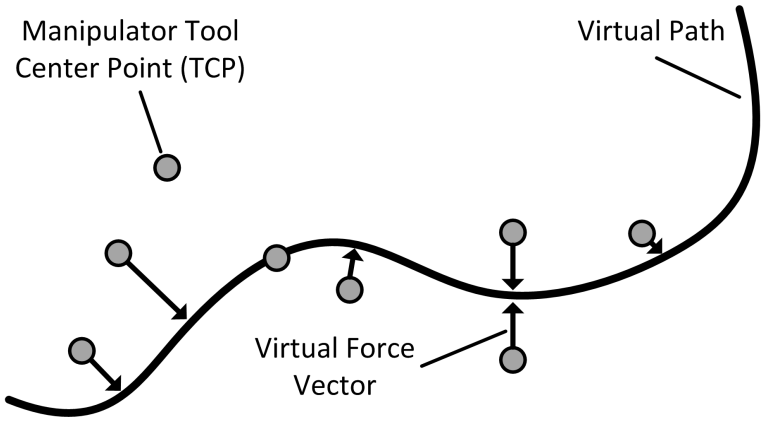


Figure 3.8: a 2D virtual path.

The admittance type GVFs do not generate force on their own, but rather guide the force that the operator exerts to the system. E.g. an admittance fixture can apply friction if the operator tries to move in to an undesired direction. The admittance control is typically implemented to follow the equation:

$$v = K_f f, \tag{3.12}$$

where v is the output velocity vector, K_f is an admittance gain matrix and f is the force applied by the operator. Benefit of the admittance GVF is its passivity. The slave velocity is always proportional to the force applied by the operator and therefore the manipulator can not move without the operator exerting force to the system [2]. An admittance GVF can also be either soft or hard. A hard fixture means that no movement of the manipulator is allowed at all and soft means that the manipulator can be moved to an undesired direction but the operator has to fight the manipulator in order to do that. Disadvantage of the admittance GVF, and the admittance virtual fixtures in general, may be that slow operator drifting in to the undesired area is inevitable even if the user has no such intentions [18].

The problem of the impedance GVFs is that they are active fixtures. Therefore, the fixture can generate force without an operator interaction. The stored energy can unintentionally move the master manipulator, generating potentially dangerous glitch in the position of the slave manipulator. Because of this danger, the admittance GVFs are generally safer and therefore more attractive choice than the impedance GVFs. However, the admittance virtual fixtures are impossible to directly implement on the impedance systems. Teleoperation systems usually always are impedance systems where the master is an impedance device and the slave is either

an admittance or an impedance device [2]. Few viable approaches for implementing admittance GVFs on an impedance teleoperation system have been proposed. The pseudo-admittance control by Abbot [1] is probably the most renown of these.

3.3.2 Force Generation

The force generation of an impedance type guiding virtual fixture is a straightforward application of the Hooke's law:

$$F_{GVF} = \begin{cases} 0, & x_m \geq x_o \\ K_{GVF} x_{i,} & x_m < x_o, \end{cases} \tag{3.13}$$

where: F_{GVF} is the force generated by the virtual fixture, K_{GVF} is the spring constant of the fixture, x_i is the distance vector from the closest point of the path, x_m is the shortest distance between the manipulator TCP and the path and x_o is the range of the virtual fixture. The guiding virtual fixtures are meant for attracting the TCP, therefore similar stability problems as the virtual wall contact jitter issue are not encountered with the paths and damping is unnecessary. Admittance virtual fixtures can be implemented using various sets of rules. The most common implementation technique follows the rule presented in the equation 3.12.

The bilateral control system of the DTP2 is an impedance-type telemanipulation system. Therefore it was concluded that the focus for guiding path implementation should be in the impedance type paths. The manipulators have several levels of safety systems that prevent the manipulator from doing fast and unexpected movements. Safety features of the manipulator were considered to be adequate for covering the potential issues with the activity of the impedance type GVFs. Also an impedance control scheme for the manipulators has been previously developed, allowing the usage of the admittance GVFs. However, the developed impedance control has limitations that make it difficult to use in some situations.

4. DTP2 COMPUTER-AIDED BILATERAL TELEOPERATION CONTROL SYSTEM

Remote handling (RH) control systems and manipulators for the ITER divertor maintenance are the main research field of the DTP2, located at Tampere, Finland. At the moment, the DTP2 houses two prototype robotic manipulators that are used for the divertor maintenance research. These are the WHMAN, which is a 6 DOF manipulator, composed of a robotic arm and a spherical wrist, and the Cassette Multifunctional Mover (CMM). The WHMAN is designed for delicate remote maintenance operations, while the CMM is a heavy lifting 3 DOF manipulator for operations that require very high payloads.

The remote handling manipulators of the DTP2 are controlled with a prototype remote handling control system (RHCS). The RHCS provides a large array of tools and control modes for the operators. One of these tools is the CAT system, which is developed in this thesis. CAT implements the haptic shared control mode of the RHCS. The bilateral and shared control systems introduced in this thesis were developed for the WHMAN but are generic and can be used for other manipulators as well.

This implementation chapter first briefly introduces the bilateral teleoperation implementation of the WHMAN and shows the integration of the CAT subsystem into it. Later the design and detailed implementation of CAT is introduced. The next chapter introduces test results gained with the CAT system in a teleoperation experiment. The purpose of the experiment is to prove the feasibility of the haptic assistance in the remote maintenance tasks of the DTP2 environment.

4.1 Bilateral Teleoperation Control System

The DTP2 control system provides multimodal feedback for the operator team of the maintenance manipulators. Available feedback methods are visual feedbacks in form of virtual models or video, audible feedback and haptic feedback. The focus at the DTP2 has been on the usage of virtual models and haptic feedback, because the ITER environment will severely limit the quality of direct camera viewing [14]. Providing the video feed from the ITER reactor will be difficult for a multitude of reasons. The high residual radiation level of a stopped fusion reactor damages conventional electronics, therefore only radiation tolerant cameras can be used. All

the surfaces of the environment are metallic creating reflections and poor contrast. Also, the space available for camera installation inside the divertor maintenance tunnel is limited reducing the achievable field of view.

The DTP2 RHCS is a heterogeneous distributed software system that consists of several networks and applications. The heterogeneity of the technologies originates from the fact that the requirements for the different software subsystems are mixed favouring different solutions for different subsystems. Also some legacy code and hardware has reduced the possibility of freely choosing solutions. This section briefly describes the manipulators, haptic shared control implementation and bilateral teleoperation control system of the WHMAN.

4.1.1 DTP2 Software Architecture

The DTP2 RHCS architecture is an adaptation of the ITER RHCS architecture, proposed by Hamilton in [14]. Hamilton's architecture is presented in the Appendix 1. Figure 4.1 presents the DTP2 control system architecture.

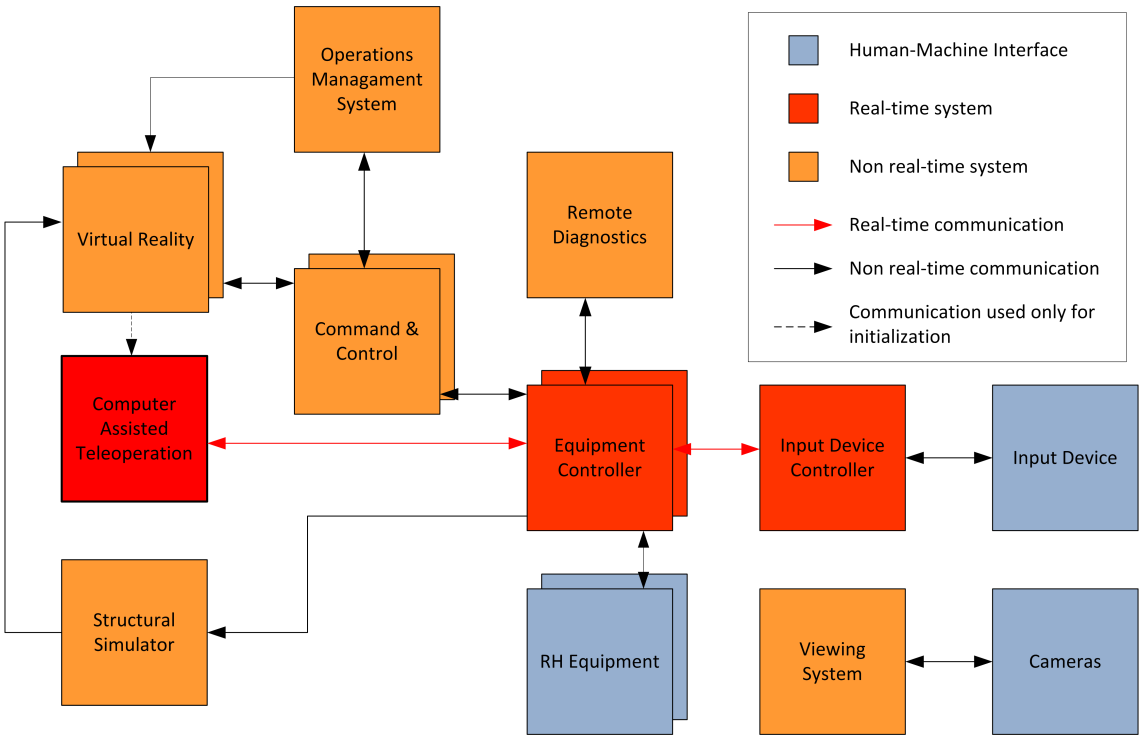


Figure 4.1: DTP2 RHCS (Remote Handling Control System) top-level architecture. Subsystems with red background or rim are real-time systems and communications between these subsystems have real-time requirements. [3]

The parts of the control system that are related to the bilateral control of the manipulators are:

- Command & Control (C&C),

- Equipment Controller (EC),
- Input Device Controller (IDC),
- Input Device (ID),
- Computer Assisted Teleoperation (CAT),
- Virtual Reality (VR).

Figure 4.2 depicts the participant subsystems of the bilateral teleoperation implementation in more detail.

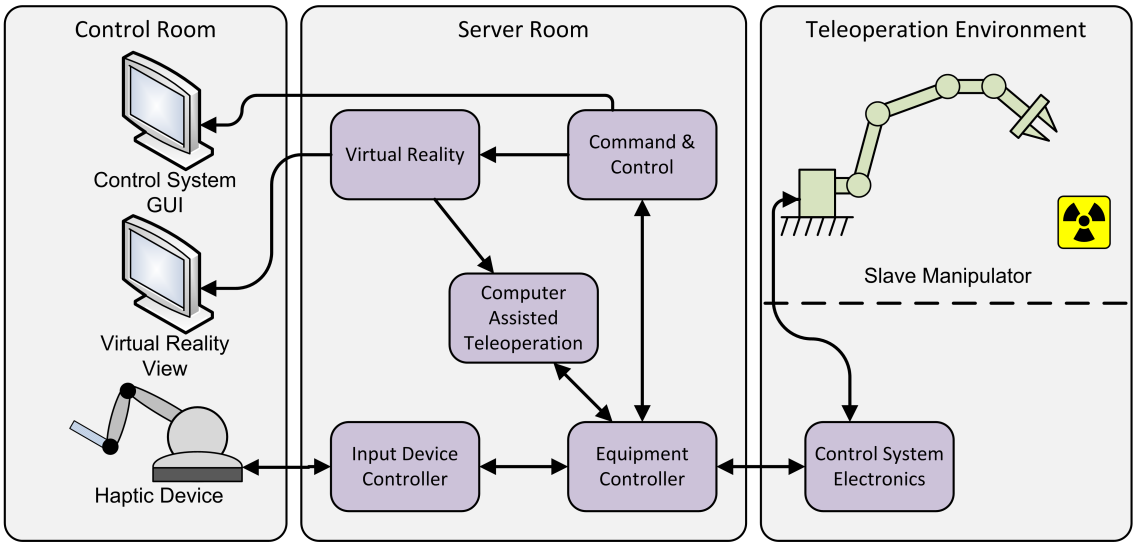


Figure 4.2: DTP2 high level bilateral teleoperation control system architecture.

The control system implements a master-slave bilateral teleoperation scheme, where a commercial haptic device is used as the master and the WHMAN as the slave manipulator. The C&C subsystem provides the graphical user interface for the RH operators. The VR subsystem used in the DTP2 RHCS is IHA3D visualization software, which has been developed at IHA for this purpose [24]. The VR subsystem provides an on-line view of the virtual models and the teleoperation environment. The VR system is also used for setting parameters for the CAT system. CAT generates the assisting virtual forces that are reflected to the master side of the bilateral teleoperation system.

CAT, IDC and EC are all real-time systems, executed at a 1 kHz frequency. CAT and the IDC are based on Linux operating systems (OS) and The EC is developed for the LabVIEW Real-Time Module-OS. The communication links between the subsystems are implemented using different techniques. Particularly all the interfaces of the CAT system are implemented using the DDS (Data Distribution Service) middleware. CAT also offers a basic UDP as an alternative interface.

4.1.2 Slave Device

The WHMAN is a 6 DOF manipulator, composed of a robotic arm (with three rotational joints and one prismatic joint) and a spherical wrist (three rotational joints) that is attached to the end of the arm. The whole manipulator is installed on top of a linear joint that is mounted on the CMM device. All the joints are used continuously and concurrently, so the manipulator achieves redundancy with eight active joints. A 6 DOF force sensor is attached to the tip of the manipulator. This allows contact force/torque measurements that are used for the haptic feedback. Figure 4.3 shows the WHMAN.

Figure 4.3: WHMAN without the prismatic joint installed under the manipulator arm.

The WHMAN was developed specifically for ITER divertor maintenance tasks. Dimensional and performance requirements of the manipulator were based on analysing different RH scenarios. Hydraulic actuators were chosen as power sources to ensure sufficient torque. The addition of a prismatic joint allowed extending the work envelope, while still being able to manoeuvre in tighter spaces than without it. To minimize space requirements, the manipulator can also be folded into a storage configuration.

Demineralized water was chosen for power transmission of the WHMAN because the usage of oil hydraulics is not allowed in a fusion reactor [10]. Water does not get activated by the radioactivity and water spills are easy to vaporize completely, both of which are attractive features for the ITER maintenance. Another special feature of the manipulator, making it suitable for ITER conditions, is the usage of only analog sensors and actuators. Active electronic components are practically impossible to use with the ITER RH manipulators because the radiation quickly damages normal semiconductors. Commercial off-the-shelf active electronic components with sufficient radiation tolerance ratings are very rare and radiation shielding of standard industrial components is not practically possible due to space limitations. Conceptual and engineering design of the manipulator, including the water hydraulic stainless steel vane actuators, has been carried out in the Department of Intelligent Hydraulics and Automation [32].

4.1.3 Master Device

The master device used with the WHMAN bilateral teleoperation control system is a Phantom Premium 3.0 6DOF-haptic device, manufactured by Geomagic-Sensable. The device provides force feedback in three translational degrees of freedom and torque feedback in three rotational degrees of freedom. Hence, the operator can feel, not only the Cartesian contact forces from the environment, but also the associated rotational torques. Table 4.1 summarizes some of the main characteristics of the Phantom Premium. Figure 4.4 shows the Phantom Premium.

Figure 4.4: Phantom Premium 3.0 6DOF-haptic device [13].

Table 4.1: Phantom Premium 3.0 6DOF technical specification [13].

Workspace	Translational	838x584x406 mm
	Yaw	297^{o}
	Pitch	260^{o}
	Roll	335^{o}
Nominal Resolution	Translational	Approx. 0.02 mm
	Yaw & Pitch	0.0023^{o}
	Roll	0.0080^{o}
Backdrive friction	Translational	0.2 N
Maximum exertable force	0.2N	
Maximum exertable torque	Yaw & Pitch	188 mNm
	Roll	48 mNm
Stiffness	$1\frac{N}{mm}$	

A single push button is available on the haptic device handle for re-indexing. When the button is not pressed, the master manipulator is disengaged from the slave and can be repositioned independently. Pressing and holding the push button engages the master with the slave and the operator can start controlling the motion. The Phantom device also electrically recognizes the presence of the operators hand and disables force feedback if the hand is not present.

4.1.4 Bilateral Teleoperation Implementation

The WHMAN control system supports a number of different manipulator control methods. These include: the direct control of joint actuators and controllers using a joystick or a keyboard, the automatic trajectories for cartesian and joint space, and the bilateral teleoperation utilizing a haptic device. Most of the WHMAN teleoperation tasks are delicate operations relying heavily on the operator intuition and decision making. Therefore, most of the WHMAN operations are done in bilateral teleoperation mode. At the moment, both the 4-channel, and the traditional P-F architectures have been implemented for the WHMAN. The control mode can be changed during run time.

There is some kinematic similarity between the haptic device used as the master and the WHMAN, but these devices are far from identical. The master device is an impedance device with relatively small workspace and low force output whereas the WHMAN is a large admittance device and has to be able to apply significant forces during the maintenance tasks. Therefore, movements and forces of the bilateral teleoperation system are heavily scaled and the system includes a gravity compensation function for the tools of the manipulator. Force scaling is necessary

for the WHMAN to reduce the physical strain of the operator to a reasonable level. The drawback of the scaling is that it diminishes the transparency of the teleoperation system. Tuning of the bilateral teleoperation control system has been aimed to provide a good balance between transparency and operator comfort.

4.2 CAT Design and Implementation

The CAT subsystem is the part of the DTP2 RHCS that is dedicated for assisting the RH operators in bilateral teleoperation tasks. The operational principle of CAT is to generate virtual forces based on the virtual models of the teleoperation environment and sensor data from the RH devices. Generated virtual forces are overlaid on top of real sensor information from the slave manipulator, as proposed in [27]. The DTP2 CAT adds the artificial forces at the master side of the bilateral teleoperation system.

CAT implements two different kinds of assistance functions (virtual fixtures). These are: the point cloud[1] based guiding virtual paths and OBB based resisting virtual walls. Paths are used for guiding the operator of the haptic device along pre-set paths in the environment. Walls are mainly used for preventing unintended collisions with the environment, but can also be used for guiding purposes as in [27]. Another major part of the CAT system is the communications with other sub-systems.

4.2.1 System Analysis

The requirements for CAT were extracted and extrapolated from the technical specification document for the Grant F4E-GRT-143 [9]. The goal of the requirement analysis was to match the requirements set by the control system architecture and research goals set for the subsystem. Summary of the requirements analysis results is presented in [3] and in Table 4.2.

[1]Point cloud is a set of points in three-dimensional space. Points can for example form a representation of a surface as in the 3D-imaging devices or paths as in the DTP2 CAT system.

Table 4.2: Requirements of DTP2 CAT, extracted from [3].

ID:	Name:	Description:
CAT_NF_REQ1	Update rate.	Update rate of virtual forces created by CAT shall be at least 500-1000HZ.
CAT_NF_REQ2	Force maximum range.	The operator shall be able to set the maximum range of GVFs. Virtual forces are only generated if the master device is in the appropriate range of a fixture.
CAT_F_REQ1	Path following.	CAT shall be able to generate virtual forces guiding the operator along a GVF path.
CAT_F_REQ2	Virtual walls.	CAT shall be able to generate resisting virtual forces when master device penetrates a pre-defined virtual wall.
CAT_F_REQ3	Enable/disable.	The operator shall be able to enable or disable CAT generated forces from the VR. This can be done individually to each virtual wall/GVF or to all the forces at the same time.
CAT_C_REQ1	Send calculated virtual force.	CAT shall send information of CAT generated virtual force to EC.
CAT_C_REQ2	Receive RH-equipment position data.	CAT shall receive data defining position of RH equipment from EC.
CAT_C_REQ3	Receive configuration data.	CAT shall receive coordinates and configuration parameters of virtual walls and GVFs from the VR.

The requirements analysis of CAT was fairly simple because the subsystem is intended to be a simple application and does not have direct interaction with the bilateral control system operators. This is reflected in the small number of requirements that, on the other hand, are demanding to achieve and leave a lot of responsibility for the developer in the decision making. The high level architecture of CAT was refined from the requirement analysis data and is presented in the Figure 4.5.

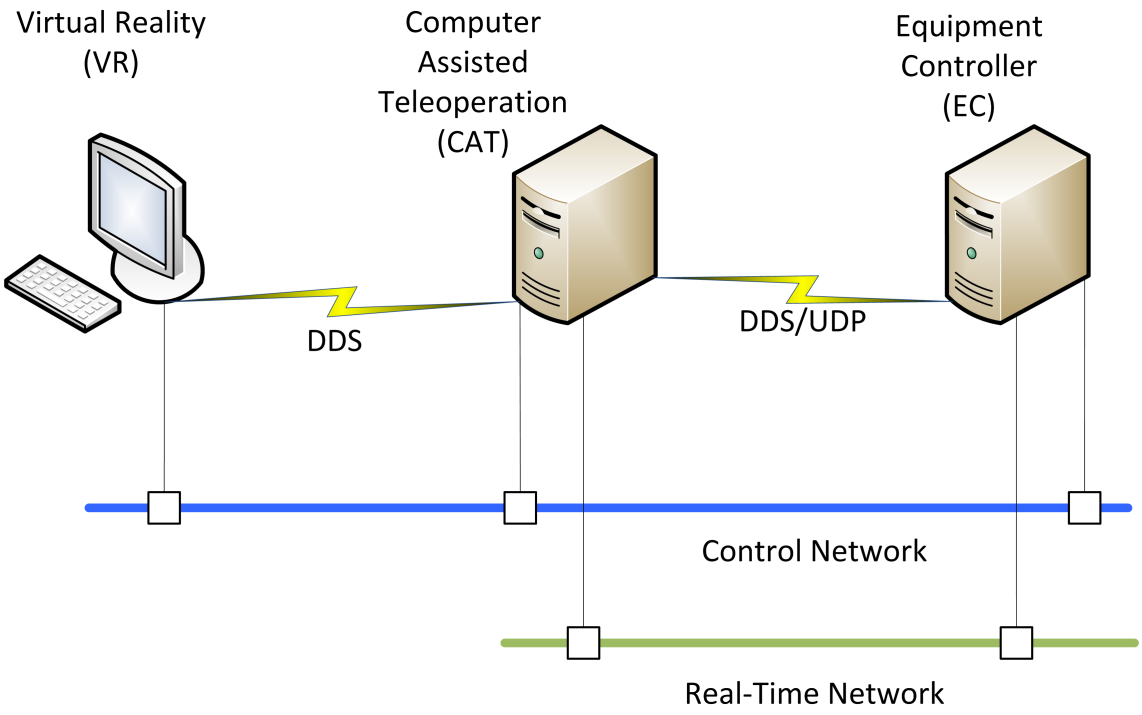


Figure 4.5: CAT related system architecture of the DTP2 RHCS.

CAT has only three outside actors affecting the subsystem. These are the virtual reality (VR), equipment controller (EC) and, to much smaller extent, the RH operator team. The VR subsystem is primarily responsible for providing the up-to date virtual models from the teleoperation environment to the RH operators. It is also used for setting virtual fixtures in the virtual environment and delivering these to CAT. The EC subsystem is the low level controller of the WHMAN. It is also responsible for providing sensory information to CAT and relaying the virtual forces to the input device controller. Neither the EC nor the VR share hardware with CAT. Communication with these subsystems happens through the control and real-time networks which are implemented using Ethernet techniques and the DDS (Data Distribution System) middleware. These communication links are introduced in more detail in 4.2.4.

The operator team can affect CAT only by applying necessary calibration offsets and turning the application on and off. During runtime, all user interaction with the CAT application is ignored apart from the shut down signal.

The CAT subsystem is implemented on an industrial PC, which is dedicated for CAT. The CPU (Central Processing Unit) of the computer is a single core Pentium 4 with 2.4 GHz clock speed. The computer also has 2 gigabytes of RAM (Random Access Memory) and a 320 gigabyte hard disk drive. Also a network adapter is required for the communications. Any other computer accessories or components are not required. The software environment used in the development of the application

is introduced in detail in 4.2.3.

4.2.2 Object Analysis

CAT is developed using object oriented methods and C++. Therefore, an object analysis was performed. The object analysis resulted to the internal division of CAT into three separate real-time tasks (threads). All the real-time tasks are autonomous, periodic and run in parallel to each other. The division is done for the sake of timing in the different components of the application. The three real-time tasks are: the controller, communication and UDP to DDS-filter. All of these tasks are operating at a 1 kHz frequency. The real-time tasks are spawned and shut down by a non-real-time main program. Figure 4.6 presents the relations between the components of CAT and associated subsystems.

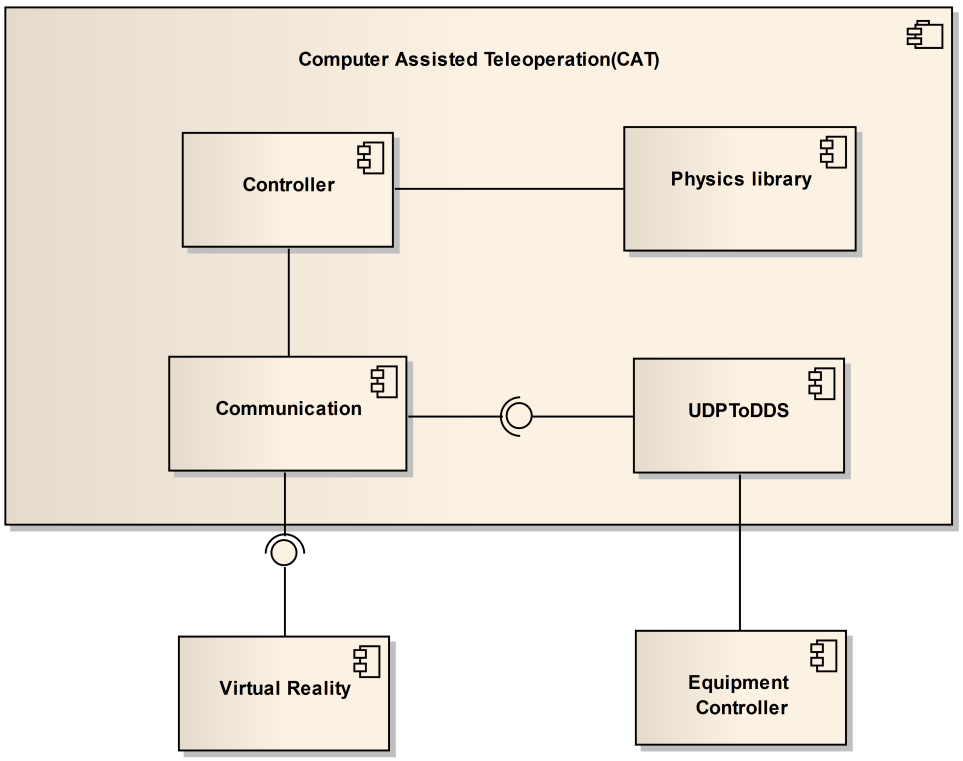


Figure 4.6: An implementation diagram showing the relations between the CAT components. Notation: UML.

The controller component of CAT is responsible for the virtual force generation based on the information that the communication task provides. The controller also uses a physics library for detecting collisions between virtual walls. Both the virtual wall and path features are implemented in the same task. Figure 4.7 presents the conceptual activity of the controller task in one scheduled 1 ms period.

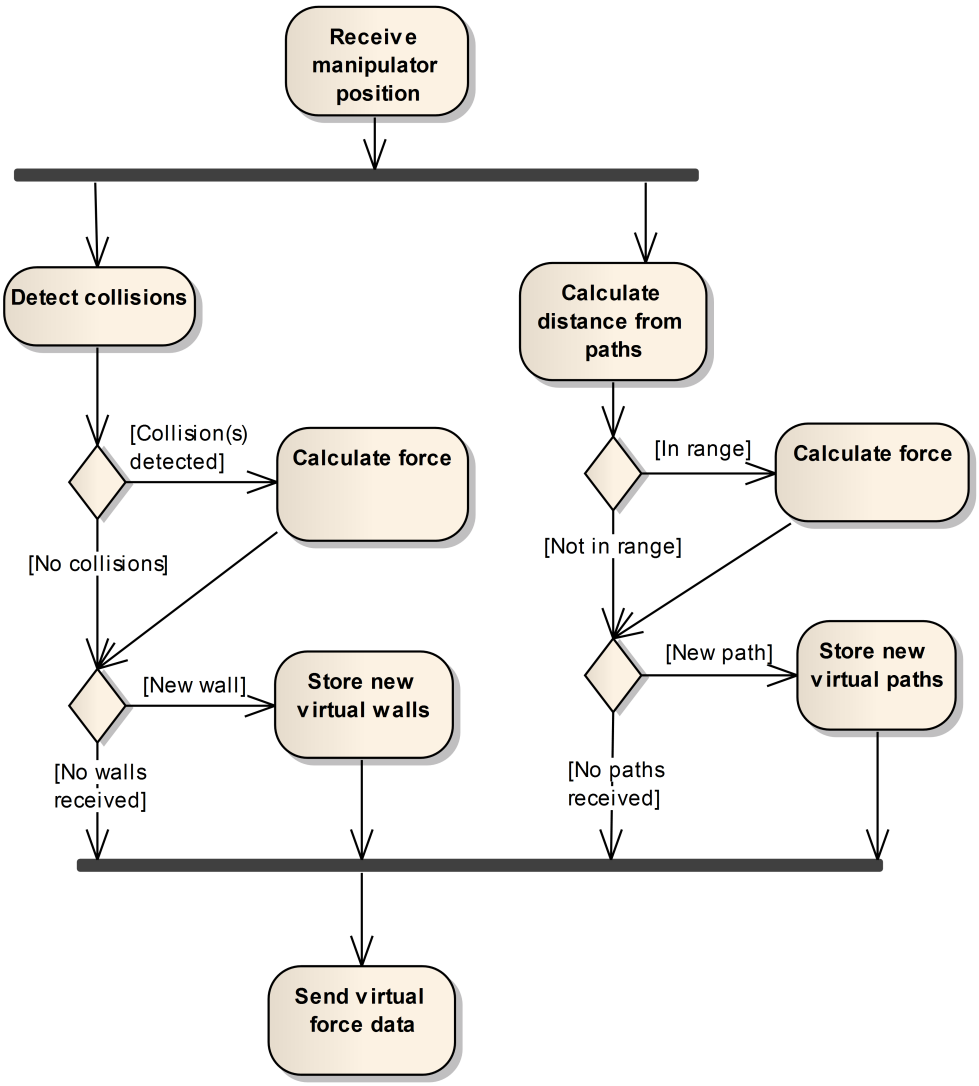


Figure 4.7: An activity diagram of a single controller task iteration.

The communication component implements the DDS interfaces of the CAT system and feeds information about the manipulator position to the controller task. The communication component is also responsible for the initialization of the inter-thread communication methods.

The UDP to DDS filter is an optional task of CAT. The component simply listens to incoming data from the EC if it is in standard UDP form, and publishes the same data in the DDS form. The same filter also listens to virtual forces published through DDS and transforms them to standard UDP. This kind of filtering had to be used with the WHMAN because the EC is developed using LabVIEW Real-Time Module. At the time of the CAT development no DDS implementations were available for the LabVIEW Real-Time Module.

The controller and communication tasks share an address space for shared memory

communication. Thread safety in that address space is guarded with mutexes[2]. The UDP to DDS filter task communicates with the communication task through the DDS middleware. DDS also uses shared memory for the inter-thread communication but the implementation for thread safety is abstracted inside the DDS implementation. All three real-time tasks and the main program also use a real-time message queue for notifying when new virtual walls or paths are available in the shared memory and for the shut down sequence of the application.

As can be seen from Figure 4.6, and from the fact that CAT consists of several threads, CAT loosely follows a layered architectural pattern with vertical slicing. CAT is a part of a bilateral teleoperation system which are notoriously intolerant for delays without additional compensations and therefore determinism is required from CAT. The formal requirement for timeliness is a firm real-time requirement. A firm deadline is a combination of shorter soft requirement and a longer hard requirement [7]. Therefore, CAT should meet the soft deadline every time with all the threads but missing an occasional deadline is not a critical problem for the safety of teleoperation. In case CAT misses a communication deadline the system renders the same force as at the previous deadline. Therefore CAT is tolerant to single random package losses.

4.2.3 Development Environment

The CAT application is developed using only open source software. The real-time (RT) constraints of CAT were a major concern for the design and development process. The solution was to use a Debian Linux operating system enhanced with the Xenomai real-time kernel extension. In this configuration the system runs as a dual kernel system where the Xenomai works with the Linux kernel and provides hard real-time support for the user space Linux programs using this feature. The three real-time tasks of CAT are using the Xenomai core extension for scheduling and inter-thread communications. The main program of CAT stays in the Linux user space on a low priority while the RT tasks are running. The Xenomai API supports several widely used RT skins, e.g. VxWorks and Posix. The native Xenomai skin was used for CAT.

The communications of CAT were implemented using the OpenSplice DDS middleware implementation, developed by Prismtech. Prismtech produces an open source community and a commercial version of the implementation. The commercial version offers more features than the community edition but basic functionality is the same in both versions. The community version of OpenSplice, distributed under

[2]Mutex (Mutual Exclusion) in software engineering terms refers to the problem of ensuring that only one process or thread can be processing a single memory section at a time. In this context mutex means the algorithm, which is used for solving the mutual exclusion issue.

the LGPL-license, was used for CAT. The Linux version of the community edition and the C++ API were used for CAT.

An open source physics library, called the Open Dynamics Engine (ODE), was used for the collision detection of CAT. ODE is a community developed library that is distributed under the LGPL license. ODE can be used with most common operating systems. Table 4.3 and Table 4.4 summarize the external libraries and software development tools used in the development of the CAT subsystem.

Table 4.3: Development environment.

Component:	Version:	Purpose:
Debian Linux	6.01	Operating system
Xenomai	2.5.6	Real-Time development framework
G++	4.4.5	C++ compiler

Table 4.4: External software components used in the CAT development.

Component:	Version:	Purpose:
OpenSplice DDS	5.4.1	DDS-middleware implementation
Open Dynamics Engine	0.11.1	Physics engine

4.2.4 Interfaces and Data Content

As can be seen from the Figure 4.5, CAT is connected to the VR and EC. The communication from the VR to CAT is unidirectional and uses the control network of the DTP2 RHCS. Through this communication link CAT receives the information about virtual models of the RH-environment. Communication with the EC is bidirectional and has RT constraints. This communication link uses the RT network of the DTP2 RHCS. Through the EC communication link CAT receives information about the manipulator position and sends the virtual force information to the low level controllers. The timeliness of the EC communication link was a concern because the amount of data transferred within the network is substantial. However, it became evident during the testing that the control network was able to easily handle the amount of data. Summary of the DDS interfaces of CAT is presented in Appendix 2.

One of the defining characteristics of the DDS middleware is the usage of quality of service (QoS) settings. The QoS settings can be used for adapting the middleware technology to a wide variety of applications. In the CAT project, the communication with the EC has real-time constraints and the timeliness of the data is of paramount importance. Simultaneously the communication with the VR requires

absolute reliability but timeliness is not a factor. The DDS QoS settings of the DTP2 communications were tuned to fulfil these requirements. All DTP2 DDS QoS settings are shown in the Appendix 4.

The coordinate system origins of the VR and EC are not the same in the DTP2 RHCS. Therefore, CAT has to know the offset between these two different origins in order to ensure accuracy of force generation. The offset information is read from a configuration file at the start of the application. The actual measurement of the offset is manual labour, which at the moment cannot be automated. Another configuration information the CAT needs is the IP-address and the port of the UDP socket of the EC. The IP information is set in a configuration file as well. However this information is required only if the UDP to DDS filter has to be used because the DDS specification defines an automatic discovery functionality.

CAT does not store persistent information about the virtual model objects. Therefore this information along with the virtual fixture parameterization has to be resent if the CAT application is restarted. While the CAT program is running, the virtual fixture information is stored in the class structure of the application.

4.2.5 Virtual Paths

Impedance type virtual paths are implemented in CAT using point clouds. The points of a point cloud are connected with straight lines forming a path. The path starts generating virtual forces when the tool center point (TCP) position of the manipulator is in the range of the path. Paths do not have a direction and therefore the operator is able to move to both directions of the path with the same effort. Paths can also be entered from any point, simply by entering the range of the path with the manipulator TCP. Similarly, path guidance can be turned off by moving the manipulator TCP outside the range of the path or by turning the path off through the VR.

The virtual path force generation algorithm of CAT implements the following pseudo code:

```
CalculatePathForces(paths, tcp, output_force)
  output_force := 0
  for i:=0 to paths.length do
    if InRangeOfPath(paths[i], tcp)
      shortest_tcp_to_path := CalculateVector(paths[i], tcp)
      force_vector := CalculateForce(shortest_tcp_to_path)
      output_force := output_force + force_vector
```

The distance between the TCP and a path segment is calculated for each segment of each path. The shortest distances from the TCP to each path are always used for calculating the feedback force. The precise distance between the TCP and a path

segment is calculated as follows:

$$\bar{w} = \bar{y} - \bar{x}, \tag{4.1}$$

$$\bar{v} = \bar{p} - \bar{x}, \tag{4.2}$$

$$x_{dist} = \frac{\|\bar{v} \times \bar{w}\|}{\|\bar{w}\|}, \tag{4.3}$$

where $\bar{x}$ is the coordinate vector of the start point of a path segment and $\bar{y}$ is the end point. $\bar{p}$ is the coordinate vector to the TCP position.

When the shortest distance between the TCP position and a path is found the distance vector is calculated using an adaptation of the Pythagorean theorem:

$$\|\overrightarrow{xr}\| = \sqrt{\|\overrightarrow{px}\|^2 - \|\overrightarrow{pr}\|^2}, \tag{4.4}$$

where $\bar{r}$ is the coordinate vector to the intersection point between the normal vector of the path to the TCP position. The abovementioned formula only gives the distance from the TCP position to the line that goes through the path segment. Therefore the distances from the TCP position to the end points of a line segment have to be calculated and taken into account when defining the position vector. After the correct position vector is found the force vector is calculated using the Hooke's law (3.13). If the TCP is inside the range of several paths simultaneously, force vectors to each path are summed. Paths do not have orientation and therefore torque values are not calculated.

Internally the GVF functionality is implemented with three classes. Figure 4.8 depicts the relationship of these three classes forming the subsystem.

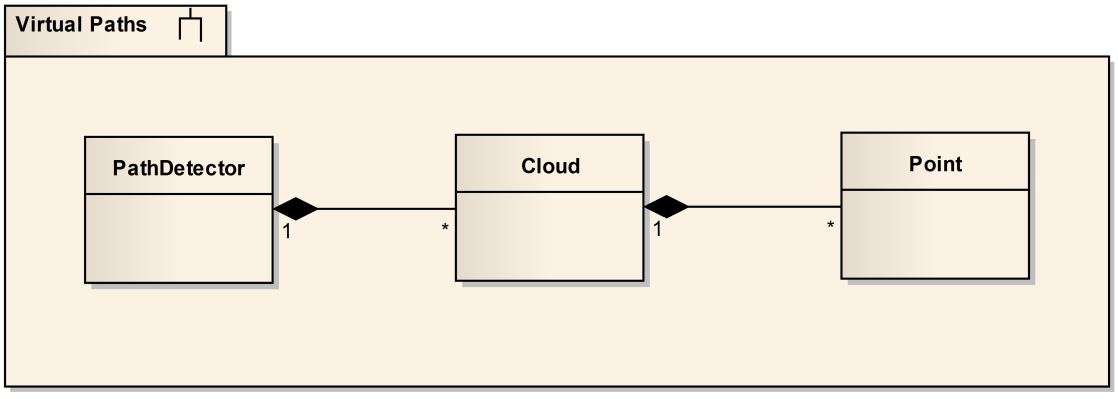


Figure 4.8: Class diagram of the subsystem responsible for force generation of guiding virtual paths. Notation: UML.

As all the primary functionalities of CAT, also the virtual path force generation is an iterative function. The function is run at a 1 kHz frequency. At each iteration, the

subsystem goes through every point cloud that is stored in to the working memory of CAT and is enabled. Force vectors of each cloud are generated separately and in the end of the iteration all generated force vectors are summed together.

In the CAT application a virtual path is defined by the coordinate sequence of the path, ID-number, enabled-flag, spring stiffness and range. The data structure is same for both, the DDS communication mechanisms and the application. The data structure of the communication channel with the VR can be seen in the Appendix 3. The message definition is in the struct called "PointCloud" and is presented in interface definition language (IDL).

4.2.6 Collision Detection

The collision detection algorithm of CAT is integrated into the virtual wall force calculation algorithms. The collision detection is easily the most computation intensive operation that CAT performs as it is calculated at a rate of 1 kHz with the rest of the force generation algorithms. The implementation is based on the standard ODE physics library which is widely used in robotics simulation applications and known for being efficient.

Collision detection performance was extensively tested during the development process. The chosen design decision is to use a simplified representation of the teleoperation environment virtual models for the force calculation. This means that only the necessary bounding boxes are present in the collision models. Also the possible bounding box shapes are limited to oriented bounding boxes. Bounding boxes for CAT are set around virtual models in the VR environment of the DTP2 RHCS. The boxes need to be manually set, although the VR system assists in the setting operation. Bounding boxes of CAT are meant to be set in the virtual model when the planning phase of a teleoperation task is undergoing but the bounding boxes can be modified or set any time during operations. Figure 4.9 shows the IHA3D interface that is used for generating CAT bounding boxes around virtual models.

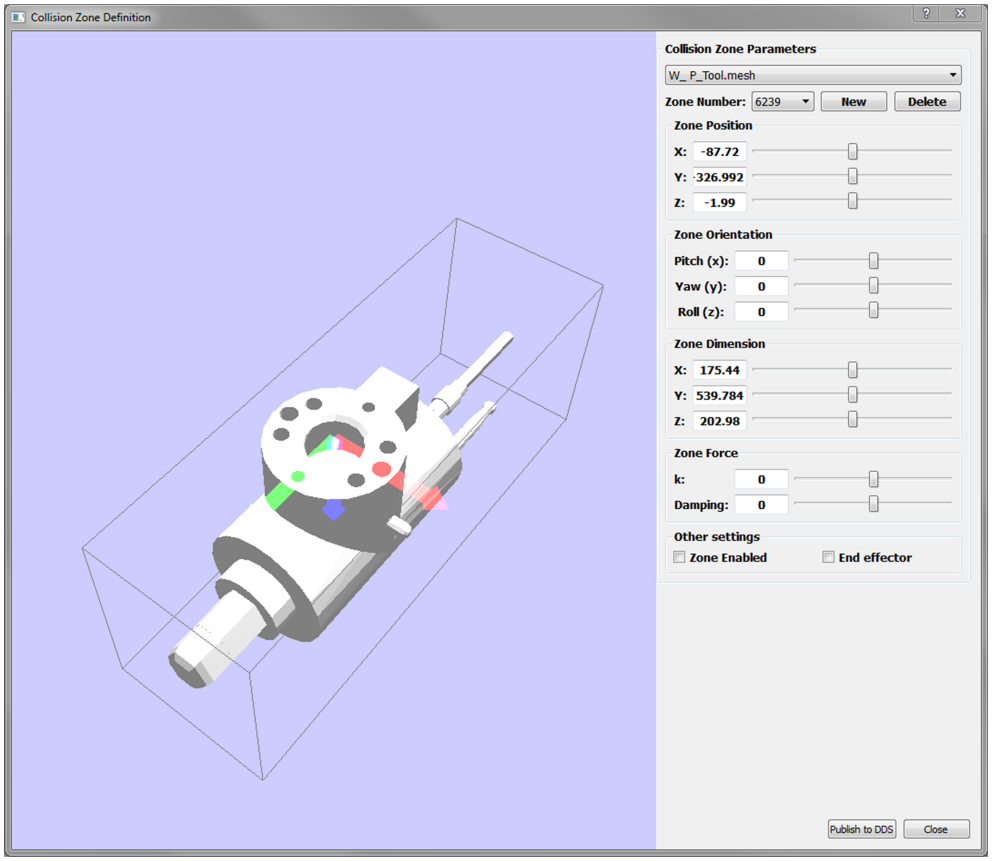


Figure 4.9: Bounding bow generation interface of the IHA3D.

With the low amount of simple bounding boxes the collision detection algorithm of CAT can be executed at a high frequency allowing realistic feeling in the user interactions between the RH operator and the virtual walls. This implementation works sufficiently for the DTP2 RHCS needs and within the timing constraints. If the calculation time would increase over the limits of the hardware with this approach, separate optimization methods could be used as in [31].

Other considered collision modelling designs were based on simulating the collisions with the normal virtual models of the VR subsystem. However, the collision detection with a high update rate and a complex model like this, would have required significantly more powerful computers than those available for this project. Another plausible solution would be to detect collisions with complex models at a low update rate (e.g. 10 Hz) and to simulate the forces induced by the collisions at a much higher rate. This approach would possibly produce excellent results, but introduces much more complexity to the system.

4.2.7 Virtual Walls

Calculating the interaction between the real world manipulator and virtual world virtual walls is a fairly complex task. Even CAT, which is designed to be efficient

and simple where possible, includes significant amount of repeated calculations for each iteration of the real-time application. The simplified functionality of the force generation of the virtual walls is presented in the following pseudo code:

```
CalculateWallForces(tcp, end_effectors, walls)
  output_force := 0
  for i:=0 to end_effectors.length do
    for j:=0 to walls.length do
      if DetectCollision(end_effectors[i], walls[j])
        intrusion_vector := CalculateVector(end_effectors[i],
                                            walls[j])
        output_force := output_force +
                        CalculateForce(intrusion_vector)
```

In reality the application functionality is much more complex and processing intensive. For example, the application includes a significant amount of kinematics related calculations, force generation algorithms and safety functions that are not included in the pseudo code.

Internally the structure of the CAT virtual wall functionality resembles the virtual path functionality with three similarly connected objects. Figure 4.10 depicts the extracted class structure responsible for calculation of virtual wall forces.

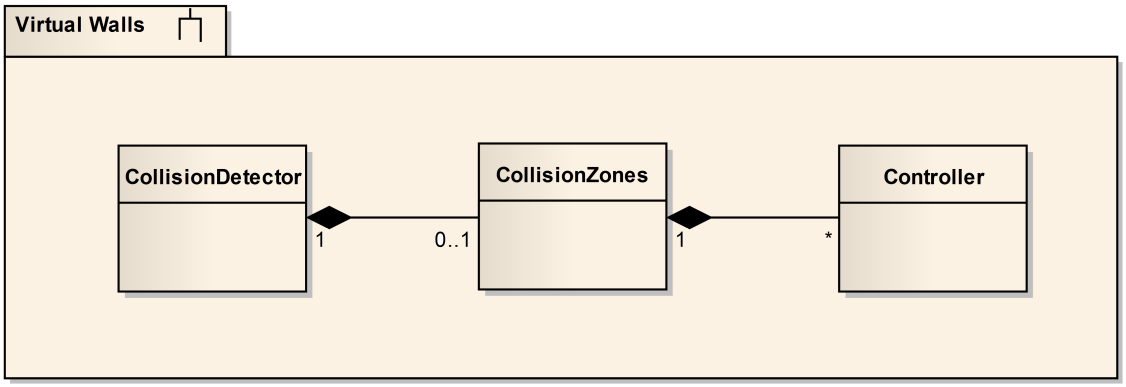


Figure 4.10: Class structure of the subsystem responsible for calculating interaction forces between virtual walls and real world manipulators. Notation: UML.

Information about the virtual walls is stored in two separate locations of the application. An ODE world contains the information about the size, position and orientation of each virtual wall in the virtual model. The same information is also stored in the controller class, together with the virtual wall ID, enable flag, spring stiffness, damping and end effector flag. Information about the end-effector position is updated at the beginning of each iteration. The virtual walls are updated only when new information is received from the VR. The data structure that is used for storing information in the controller class is similar to the structure used in the

communication protocol. The data structure used in the communication channel can be found in IDL-format in the Appendix 3.

5. DTP2 COMPUTER-AIDED TELEOPERATION EXPERIMENT

Quantitative analysis of the performance of bilateral teleoperation systems is fairly difficult, since the most significant contributing factor to the overall teleoperation performance is the operator. The capabilities of operators vary significantly. For this reason, the performance of a bilateral teleoperation system is typically evaluated using questionnaire study methods and statistical analysis. This approach was also chosen for the DTP2 CAT.

The purpose of this thesis was to investigate the implementation techniques used in the development of the DTP2 CAT and the effectiveness of the system in an ITER relevant divertor maintenance scenario. To study the effect of the haptic assistance to the bilateral teleoperation, an experiment was conducted. In the experiment, a set of test operators performed a divertor maintenance related operation with the WHMAN. During the experiment, performance of the operators was observed in terms of execution times, accuracy and operator work load.

5.1 Teleoperation Test

The divertor cassette locking sequence, which has been extensively researched at the DTP2, was analysed to determine a suitable maintenance task for the experiment. In the cassette locking sequence, the WHMAN applies preloading tension to a divertor cassette and locks it to the divertor rails using the cassette locking system (CLS) of the cassette. The whole cassette locking sequence is far too complex and long for the purpose of the CAT experiment. Therefore, a smaller segment of the task was chosen.

The most common steps of the CLS sequence are attaching to and detaching from different tools used by the WHMAN. The process is essentially a "peg-in-hole task", similar for each tool. These tasks were also seen to be simple enough for operators that had varying levels of teleoperation experience. Nevertheless, attaching to a CLS tool requires very high accuracy and adaptability due to the mechanical interface of the WHMAN tool changer. Moderate misalignments make it impossible to pick up a tool with the WHMAN tool changer. Therefore, automation of the pick-up tasks difficult to implement. Figure 5.1 illustrates the tool changer interface of WHMAN and the pin tool.

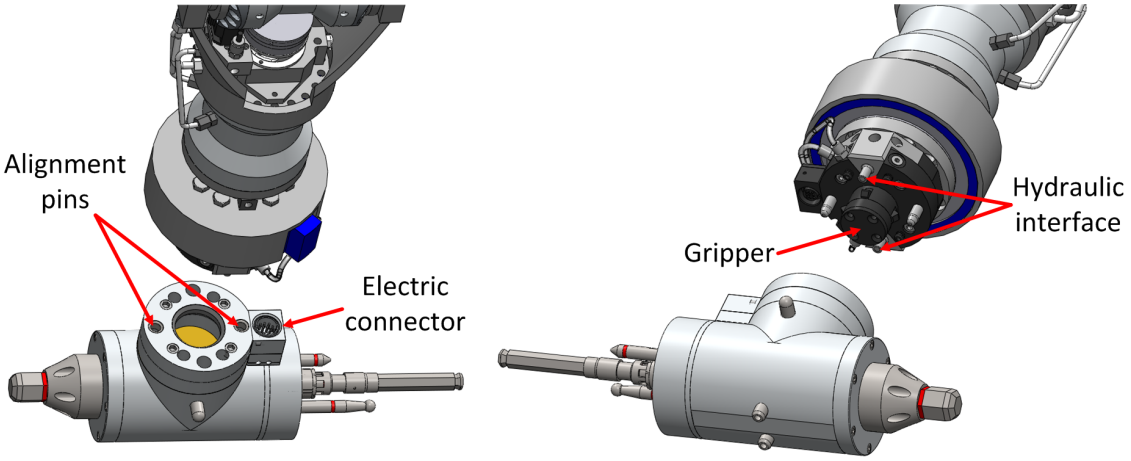


Figure 5.1: The WHMAN tool changer interface and the wrench-pin tool used in the CLS.

When picking up the wrench-pin tool, the WHMAN approaches the tool changer interface from above. The operator adjusts the orientation and location of the WHMAN with the haptic device to align the alignment pins and the electrical connector of the WHMAN. When the pins and the connector are aligned the operator inserts these inside the wrench-pin tool. Full insertion of the tool changer is verified from the video feed and the 3D model. After the tool changer is inserted, the gripper of the tool changer is operated in order to lock the tool to the manipulator. Figure 5.2 shows a simplified presentation of the pick-up task.

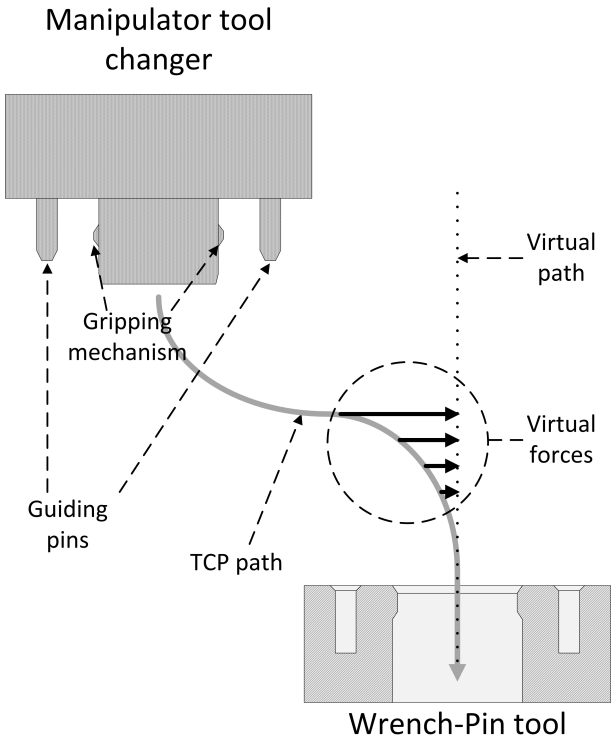


Figure 5.2: Wrench-pin tool pick-up task overview.

The tool changer of the WHMAN has two alignment pins, one locking pin, two hydraulic interfaces and an electrical connector that, in the worst case, have to be aligned for picking up a tool. The wrench-pin tool of the WHMAN was used in this teleoperation experiment because it does not require the alignment of hydraulic interfaces making it easier for the operator. To further simplify the task the orientation of the manipulator was locked in correct orientation throughout the tests.

5.2 System Configuration

The WHMAN, together with the full DTP2 RHCS, was used in the experiment. The RHCS was also used for data logging during the experiment. A Phantom Premium 3.0 6DOF commercial haptic device was used as the master device. The experiment was performed on a mock-up test stand of the manipulator. The mock-up is geometrically identical with the real divertor cassette locking mechanism. The test stand assembly and CLS are introduced in detail in [21].

Participants had direct vision of the slave manipulator during the experiment for safety reasons. However, the operators were not able to see the tip of the manipulator or task-related details directly and were specifically instructed to rely on the visualization model and a video feed from the environment. Snapshot from the virtual model of the environment and the low contrast video feedback available for the operators during the experiment are shown in Figure 5.3.

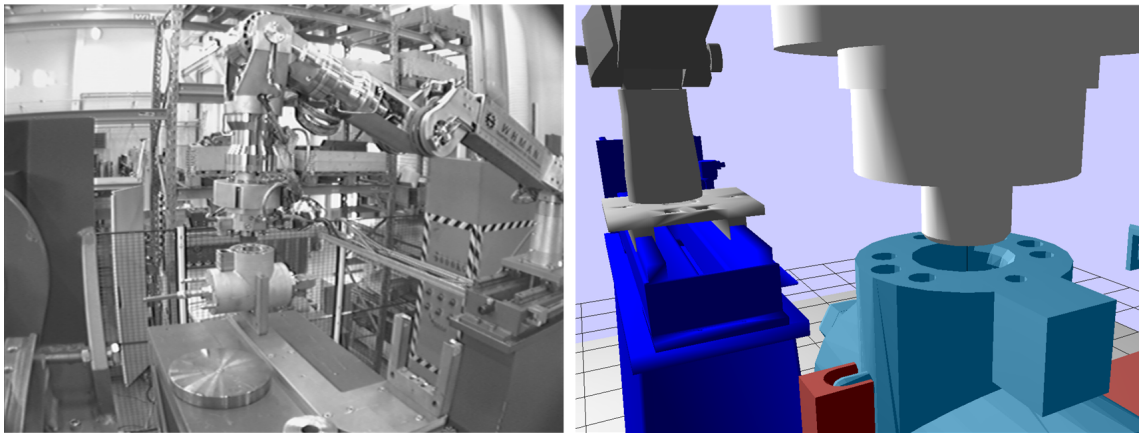


Figure 5.3: Views of the teleoperation environment available for operators during teleoperation tests. The left figure is a screen shot from the video uplink and right side is a screen capture from IHA3D visualization model. The virtual model view can be moved and can contain several windows.

The haptic guidance consisted of two virtual paths that guided the operators to the tool changer interface. These paths were linear and laid over each other. The purpose was to provide one path with a long range but a small force effect to give rough guidance to the interface. Another path had a short range but a high force

guidance effect.

Trajectories of the master and the slave devices were recorded during the teleoperation tests. The recording was done with a LabVIEW software developed for this purpose. The recording software was deployed on the EC. Trajectories were recorded at the frequency of 40 Hz which is sufficient for the slow moving manipulator.

The WHMAN control system parameters were tuned give a comfortable level of transparency to the teleoperation environment without being physically demanding to the operator of the system. CAT tests were done using the P-F controller scheme. The feedback forces were also filtered to remove some of the force jitter, originating from the signal noise of the force sensor. Parameters of the control system were identical for all the participants.

5.3 Procedure

During the experiment, 10 participants were asked to perform the pick-up operation twice with the WHMAN. One of the operations was done with the CAT system enabled, and another time without the assistance from CAT. Nine of the operators were males and one was a female. All the operators were of approximately the same age and educational background. The operators had varying haptic bilateral teleoperation experience. Some of the operators had not used a bilateral teleoperation system before but all the operators were at least familiar with the concept. Each operator was given approximately five minutes training time with the teleoperation system before the test. Altogether, the test took about 30 minutes for each operator.

Each of the pick-up operations started from the same position in the workspace. The start position was outside of the effective range of the virtual path, forcing the operators to rely on visual cues as well as the haptic guidance. The task was concluded when the manipulator was fully inserted to the tool.

The user test was organized in a repeated-measures manner. Each operator performed the pick-up task twice, once with the haptic assistance and another time without it. The two most important sources of systematic variation error when using the repeated-measure design are practice and boredom effects [11]. The systematic variation caused by these effects was compensated by counterbalancing the order in which test cases were implemented. Half of the operators did the assisted operation first and unassisted later and the other half performed the test in reverse order.

During the teleoperation tests reference and actual positions of the WHMAN TCP were recorded. The results of the teleoperation tests were evaluated in respect of three different aspects: the operator mental work load, task execution times and accuracy. Following sections exhibit the processed performance results for each evaluation aspect.

5.3.1 Task Execution Times and Accuracy

The recorded trajectories of the pick-up tasks were divided into three generic subsections for analysis purposes. These were the approach, interface search and insertion phases. The approach phase constitutes the time from initial location to the first physical contact between the wrench-pin tool and the manipulator. The interface search time was counted from the first contact to the alignment of the alignment pins. The insertion phase is the time taken from alignment to the perfect insertion of the manipulator tool changer.

Measured variance in the execution times between the participants was large and the participants reacted to the CAT system in different manners. Also, the effect of learning was significant, increasing the variance. The data between the unassisted and assisted operations are compared using paired t-tests. The measured execution time data is presented in Table 5.1 and recorded mean execution times are presented in the Figure 5.4.

Table 5.1: Execution times of the wrench-pin tool pickup.

	Approach [s]		Search [s]		Insertion [s]		Total [s]	
	No CAT	CAT	No CAT	CAT	No CAT	CAT	No CAT	CAT
Op 1	43,1	44,9	6,9	14,2	9	20,2	59	79,3
Op 2	30,3	17,1	32,9	22,5	17,2	14,6	80,4	54,2
Op 3	17,8	29,4	7,3	9,3	20,4	25,2	45,5	63,9
Op 4	31,9	34,6	22,2	12,8	9	9,3	63,1	56,7
Op 5	24,9	23,3	10,7	13,2	7,2	9,6	42,8	46,1
Op 6	25,1	31,3	9,6	5,7	6,2	18	40,9	55
Op 7	47,9	47,8	38,7	0,7	39,2	26,8	125,8	75,3
Op 8	50,7	26,5	109,8	32,8	40,3	36,1	200,8	95,4
Op 9	24,2	23,4	15,4	10,6	15	11,4	54,6	45,4
Op 10	57,3	57,2	80,1	40,1	23,5	20,9	160,9	118,2

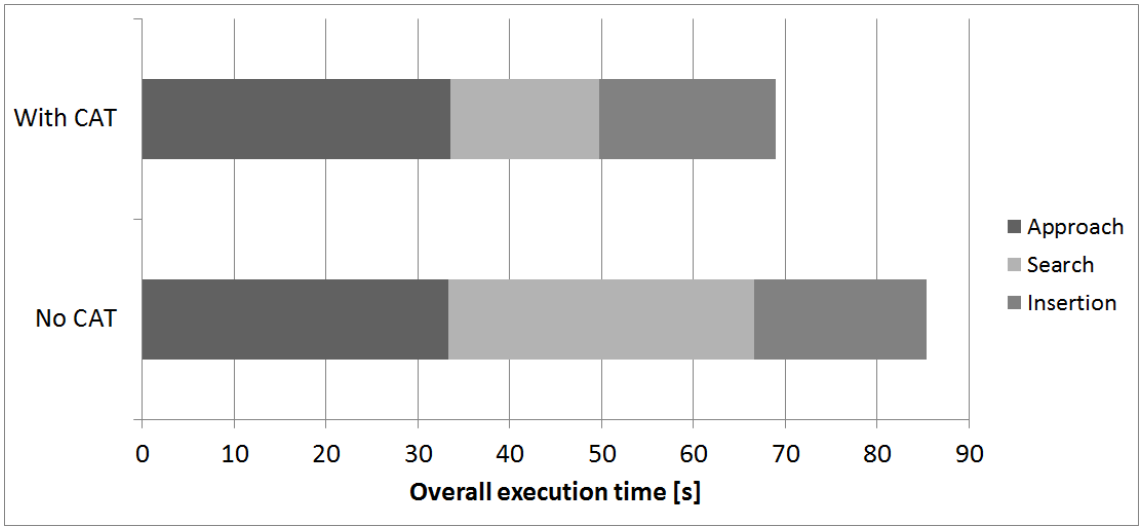


Figure 5.4: Task execution times, divided into parts according to recorded slave manipulator trajectories.

The completion time of the task improved on average by 21.1% (p=0.17) when the CAT system was in use. However, as indicated by the p-value, this result cannot be considered statistically significant. One of the operators also performed worse when CAT was enabled. Those operators who found the CAT system useful also generally performed significantly better when the CAT system was in use.

The figure above shows that the search phase, which is the most critical part of the task, was performed significantly faster (51.5%; p=0.07) with CAT than without it. As indicated by the p-value, this result is statistically stronger than the overall execution time difference but cannot be considered statistically significant either. The approach and insertion phases of the recorded tasks were performed in approximately the same mean time, both with CAT and without it.

5.3.2 Operator Workload

After each pick up operation, participants filled a Task Load Index (TLX) questionnaire form. The TLX method is a rating procedure that rates the perceived workload that the test subject experiences and divides the overall score to six subscales [23]. The test was developed in the 80s by Human Performance Group at Ames Research Centre of NASA and is widely used in teleoperation research for evaluating task load of operators.

The test produces an overall workload score and relative weights of the six workload contributing subscales. These are:

- Mental Demand,
- Physical Demand,

- Temporal Demand,
- Performance,
- Effort,
- Frustration Level.

Definition of each subscale is presented in the Appendix 5 and was also given to all the test operators before performing the TLX questionnaire.

The experimental procedure was same with each operator. First the definition sheet (Appendix 5) was presented to the operator and the operator was familiarized with the TLX process using instructional sheets of Appendix 7. After the introduction the operator performed the teleoperation task. After the completion of the task the operator was presented with 15 workload comparison cards one by one in a random order. These cards are illustrated in the Appendix 6. The operator circled the factor that contributed more to the workload in each card. Then the operator filled out the rating sheet, also illustrated in the Appendix 6. The same process was repeated after the second execution of the task.

The workload comparison cards and the rating sheet grades were used to produce overall adjusted workload ratings with the method presented in [23]. Average values among all the operators were calculated on basis of the performed TLX analyses. Results are summarized in Table 5.2 and Figure 5.5.

Table 5.2: Overall TLX scores.

Operator	No CAT	CAT
Op 1	15,3	20
Op 2	49,3	66
Op 3	20,3	45
Op 4	38,7	43,7
Op 5	21,7	29,7
Op 6	35,7	28
Op 7	17,7	44,3
Op 8	42,3	54,7
Op 9	42,7	73
Op 10	55,7	64

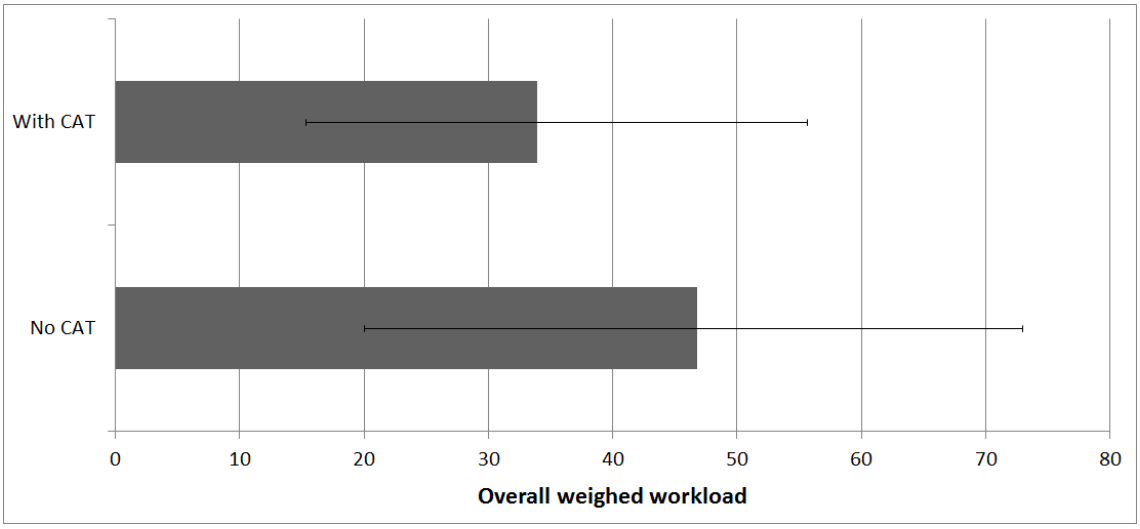


Figure 5.5: Mean values of TLX weighed workloads. The error bars represent the whole array of samples.

The TLX test revealed that the overall perceived task load was reduced by 27.5% (p=0.089) on average when CAT was used. All but one of the operators experienced a drop in the workload level. Again the variation between test subjects was large and as the p-value indicates the result can not quite be considered as statistically significant. The variance is expected because the TLX test is subjective and the participants were not given any guidelines on how to weigh the values in the answer sheet.

The detailed TLX analysis reveals that the mental demand had the largest factor contributing in the task for both test cases. Results of the detailed analysis are presented in the Figure 5.6.

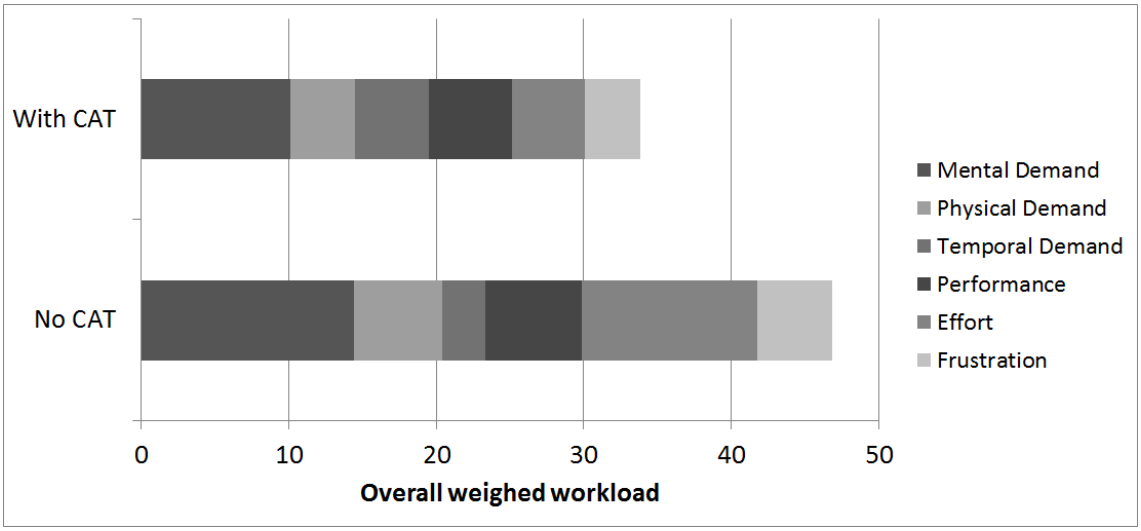


Figure 5.6: TLX perceived workload divided in to contributing factors.

When CAT was used, the required average mental demand from the operators

was 29.8% (p=0.057) lower. Another significant factor was the effort measure which was 58.3% (p=0.008) lower when operating with CAT. Temporal demand was the only measure that was relatively seen as more demanding when CAT was in use by 73.6% (p=0.015) increase. However, the temporal demand has a small weight when compared to the other factors. The results for effort and temporal demand subscales are statistically significant. The result for mental demand subscale is marginally significant.

6. CONCLUSIONS

The design and implementation process of CAT was successful. The application fulfils the requirements set for it and in general performs as expected. The application is unique in some implementation aspects; especially the collision detection running at a rate of 1 kHz with a low power computer is a significant result, demonstrating the effectiveness of the concept where the CAT algorithms are performed on a simplified representation of the virtual environment.

The most significant problems of the CAT application that were discovered during the writing of this thesis are related to the user interface of the system and usability. Offset between the coordinate systems of the IHA3D and WHMAN control systems are tedious to take into account while tuning the CAT system. Coordinate system offset does not affect the functionality of IHA3D directly because the visualization is based on joint space representation of manipulators. However, it is recommended to modify the IHA3D in future, not only for CAT purposes, but also to clear the way for other useful functionality such as measurement capability of the visualization software. Another possibility for improving the CAT system is to develop a dedicated GUI for CAT.

From the experiment point of view, the CAT haptic shared control system of the DTP2 RHCS was successful in improving the efficiency of the teleoperation system and reducing the mental workload experienced by the operator. When measuring execution times, the results achieved with CAT are well in line with similar previous research (e.g. [27, 25, 33]). The measured improvement of task execution times, particularly in the search phase of the peg-in-the-hole task, was significant and encouraging. Even thought it should be noted that the system is the first prototype version of the system and the application still has a lot of room for improvements. For example, more sophisticated fixture algorithms could be generated. Perceived task loads were also significantly lower when teleoperating with the CAT system.

It came apparent that the conducted teleoperation experiment was not thorough enough. Statistical significance would have required larger amount of data. The amount of participants was sufficient, but each participant should have repeated the task multiple times, e.g. five times with CAT and five times without it. The statistical results could have been improved further by simple measures, such as instructing the operators a little bit more about how the task was to be performed.

Nevertheless, the gained results are suggestive statistically and very similar to those that other similar research projects have produced.

It came apparent during the tests that operators have very different approaches to the teleoperation task in hand when they are not precisely asked to perform the task in a specific manner. Learning significantly affects the accuracy and the time spent for the task also. Response to assisting forces is rather individual. One of the test subjects found the assisting force disturbing and performed clearly better without the assistance, whereas the rest of the operators performed better with the assistance. Due to the fact that operators react differently to the haptic assistance, better results could possibly be gained with a more adaptive approach to the haptic guidance.

The results presented in this thesis are statistically directive, for the most part, but promising nevertheless. To get more conclusive results, more experiments need to be conducted. The pitfall of the used experimental setup is the time required for one operation. This is mostly related to the use of the WHMAN as a test device. Usage of the manipulator makes the experiment relevant for the ITER divertor maintenance, but the special-purpose design of the manipulator limits the user friendliness of the control system. The manipulator requires a certain amount of tuition to the participating operators to ensure safety of the tests.

Overall, the CAT system should be one of the focus points when the ITER RHCS, or similar systems, are developed in the future. Possibilities for increasing the efficiency and improving the safety of bilateral teleoperation with haptic shared control are simply too great to be overlooked.

REFERENCES

[1] Abbott, J. *Virtual fixtures for bilateral telemanipulation.* PhD thesis, The Johns Hopkins University, 2006.

[2] Abbott, J., Marayong, P., and Okamura, A. *Haptic Virtual Fixtures for Robot-Assisted Manipulation*, vol. 28 of *Robotics Research.* Springer Berlin / Heidelberg, 2007, pp. 49–64.

[3] Aha, L., and Saarinen, M. Implementation plan - f4e-grt-143 divertor rh updates and dtp2 phase 2 testing. Tech. Rep. F4ED23E52D v1.1, VTT/TUT, 2011.

[4] Aracil, R., Buss, M., Cobos, S., Ferre, M., Hirche, S., Kuschel, M., and Peer, A. *The Human Role in Telerobotics*, vol. 31 of *Advances in Telerobotics.* Springer Berlin / Heidelberg, 2007, ch. 1, pp. 11–24.

[5] Basañez, L., and Suárez, R. *Teleoperation.* Springer Handbook of Automation. Springer Berlin / Heidelberg, 2009, ch. 27, pp. 449–468.

[6] Colgate, J. E., Grafing, P. E., Stanley, M. C., and Schenkel, G. Implementation of stiff virtual walls in force-reflecting interfaces. In *Proceedings of Virtual Reality Annual International Symposium* (1993), IEEE, pp. 202–208.

[7] Douglass, B. P. *Real-time UML: developing efficient objects for embedded systems*, 2nd ed. Addison-Wesley, Reading (MA), 2000.

[8] Ericson, C. *Real-Time Collision Detection (The Morgan Kaufmann Series in Interactive 3-D Technology).* Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, 2004.

[9] Esqué, S. Technical specifications for divertor rh design updates and dtp2 testing (including preparation of minor upgrades). Tech. Rep. Annex B (v1.1), Fusion For Energy, 2010.

[10] Esqué, S., Mattila, J., Saarinen, H., Siuko, M., Virvalo, T., Muhammad, A., Mäkinen, H., Verho, S., Timperi, A., Järvenpää, J., Palmer, J., Irving, M., and Vilenius, M. The use of virtual prototyping and simulation in iter maintenance device development. *Fusion Engineering and Design 82*, 15-24 (October 2007), 2073–2080.

[11] Field, A. *Discovering Statistics Using SPSS*, 2nd ed. SAGE Publications Ltd., London, 2005.

[12] Geomagic. The geomagic touch haptic device, Retrieved January 9, 2014, from Geomagic: http://geomagic.com/en/products/phantom-omni/overview.

[13] Geomagic. Phantom premium haptic devices, Retrieved January 9, 2014, from Geomagic: http://geomagic.com/en/products/phantom-premium/overview.

[14] Hamilton, D. Remote handling control system design handbook. Tech. Rep. ITER D 2EGPEC v.2.3, ITER, 2011.

[15] Hannaford, B., and Okamura, A. M. *Haptics.* Springer Handbook of Robotics. Springer Berlin / Heidelberg, 2008, ch. 30, pp. 719–739.

[16] Hashtrudi-Zaad, K., and Salcudean, S. E. Analysis of control architectures for teleoperation systems with impedance/admittance master and slave manipulators. *The International Journal of Robotics Research 20*, 6 (June 2001 2001), 419–445.

[17] Hirche, S., Ferre, M., Barrio, J., Melchiorri, C., and Buss, M. *Bilateral Control Architectures for Telerobotics*, vol. 31 of *Advances in Telerobotics.* Springer Berlin / Heidelberg, 2007, ch. 10, pp. 163–176.

[18] Kikuuwe, R., Takesue, N., and Fujimoto, H. A control framework to generate nonenergy-storing virtual fixtures: Use of simulated plasticity. *IEEE Transactions on Robotics 24*, 4 (2008), 781–793.

[19] Kuang, A. B., Payandeh, S., Zheng, B., Henigman, F., and MacKenzie, C. L. Assembling virtual fixtures for guidance in training environments. In *Proceedings of 12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, HAPTICS '04.* (2004), pp. 367–374.

[20] Lawrence, D. A. Stability and transparency in bilateral teleoperation. *IEEE Transactions on Robotics and Automation 9*, 5 (1993), 624–637.

[21] Lyytikäinen, V. Development of divertor cassette locking tool prototypes according to remote handling requirements. Master's thesis, Tampere University of Technology, 2012.

[22] Mattila, J., Poutanen, J., Kekäläinen, T., Siuko, M., Palmer, J., Irving, M., and Timperi, A. The design and development of iter divertor rh equipment @ dtp2 facility. In *Proceedings of the Tenth Scandinavian International Conference on Fluid Power, Tampere, Finland* (May 2007), vol. 3, Tampere University of Technology, pp. 277–291.

[23] NASA. *Task Load Index (NASA-TLX) v. 1.0 (manual).* Human Performance Research Group, NASA Ames Research Center, Moffett Field, California.

[24] Nurminen, T. Augmented reality software module for a visualization system. Master's thesis, Tampere University of Technology, 2010.

[25] Payandeh, S., and Stanisic, Z. On application of virtual fixtures as an aid for telemanipulation and training. In *Proceedings of 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, HAPTICS 2002* (2002), pp. 18–23.

[26] Prada, R., and Payandeh, S. On study of design and implementation of virtual fixtures. *Virtual Reality 13*, 2 (2009), 117–129.

[27] Rosenberg, L. B. Virtual fixtures: Perceptual tools for telerobotic manipulation. In *Proceedings of IEEE Virtual Reality Annual International Symposium* (1993), IEEE, pp. 76–82.

[28] Salcudean, S. E., Zhu, M., Zhu, W.-H., and Hashtrudi-Zaad, K. Transparent bilateral teleoperation under position and rate control. *The International Journal of Robotics Research 19*, 12 (December 01 2000), 1185–1202.

[29] Salisbury, K., Conti, F., and Barbagli, F. Haptic rendering: introductory concepts. *Computer Graphics and Applications, IEEE 24*, 2 (2004), 24–32.

[30] Shimoga, K. B. A survey of perceptual feedback issues in dexterous telemanipulation. part ii. finger touch feedback. In *Proceedings of Virtual Reality Annual International Symposium* (1993), IEEE, pp. 271–279.

[31] Tolonen, M. Törmäystarkastelut mahdollistava 3d-visualisointijärjestelmä. Master's thesis, Tampere University of Technology, 2009.

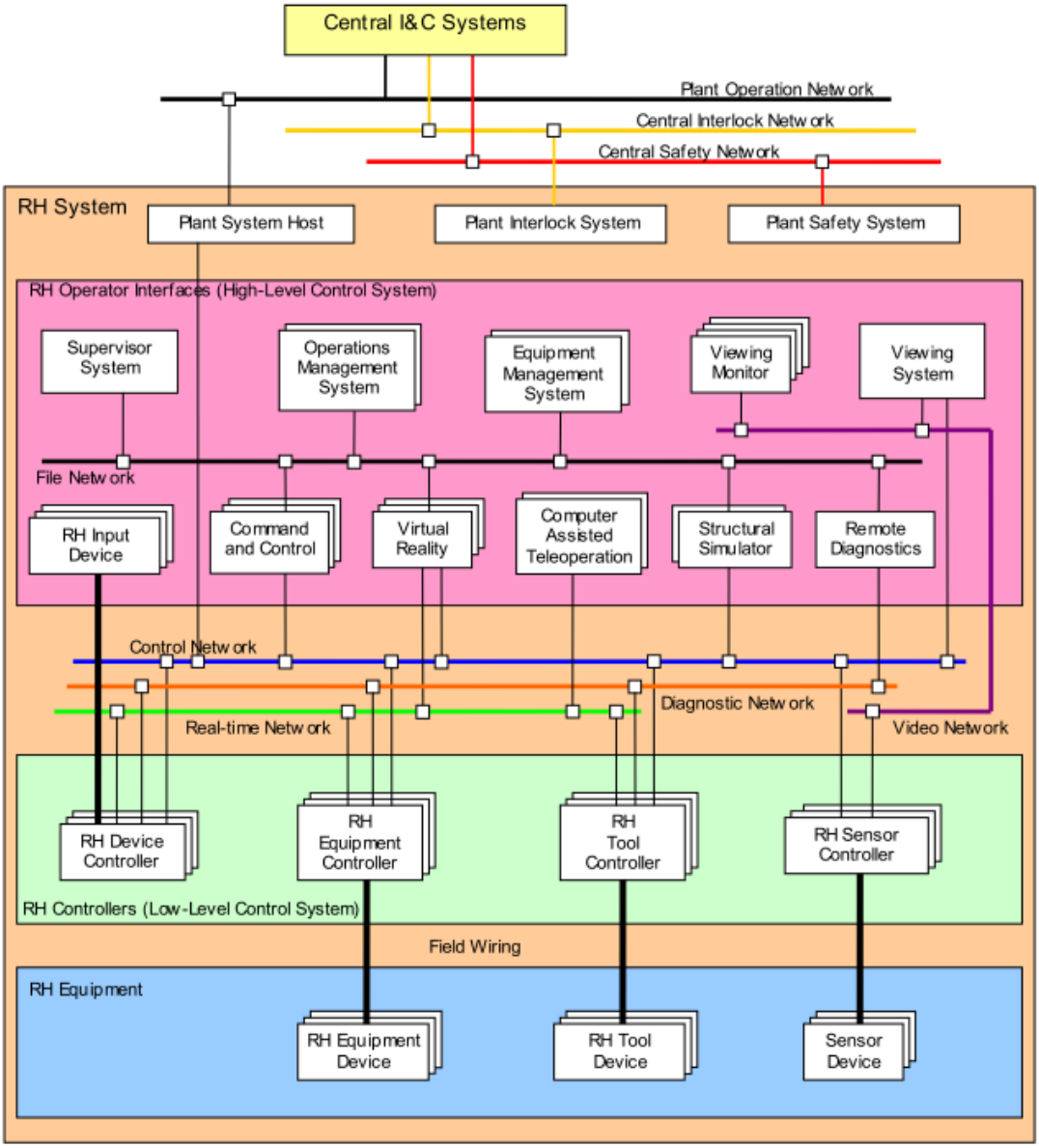
[32] Valkama, P. Design of a six degree of freedom water hydraulic arm for remote handling. Master's thesis, Tampere University of Technology, 2009.

[33] van Oosterhout, J., Abbink, D. A., Koning, J. F., Boessenkool, H., Wildenbeest, J. G. W., and Heemskerk, C. J. M. Haptic shared control improves hot cell remote handling despite controller inaccuracies. *Fusion Engineering and Design 88* (2013), 2119–2122.

[34] Vertut, J., and Coiffet, P. *Robot technology. Vol. 3B, Teleoperation and robotics : applications and technology.* Hermes, London, 1985.

[35] Zhu, M. Master-slave force-reflecting motion control of hydraulic mobile machines. Master's thesis, The University of British Columbia, 1994.

A. APPENDIX 1: TOP-LEVEL ARCHITECTURE OF THE ITER RHCS



Proposed top-level architecture of ITER remote handling control system. [14]

B. APPENDIX 2: CAT DDS INTERFACES

Publisher:		
Topic:	Description:	Data Structure:
CAT_forces	Cat generated virtual force and torque vectors in reference to the TCP of the manipulator	struct CATForce { double tcp_force[3]; double tcp_torque[3]; };

Subscriber:		
Topic:	Description:	Data Structure:
TCP_position	Position of the last static frame of manipulator, and the offset from last frame to the TCP. Both in transformation matrix form.	struct CATPosition { double transformation[16]; double tcp_offset[16]; };
Force_paths	Coordinates of virtual path way points and parameters.	struct PointCloud { long id; Location3D[] points; boolean enabled; double spring; double range; };
Force_fields	Coordinates, orientation, dimensions and parameters of virtual walls.	struct CollisionBoundary { long id; Location3D position; RotationMatrix rotation; BoxDimensions lengths; boolean enabled; double spring; double damping; boolean end_effector; };

C. APPENDIX 3: DTP2 DDS INTERFACE DEFINITIONS

```
module DTP2DDS
{
  struct CATForce
  {
    double tcp_force[3];    // x,y,z virtual forces
    double tcp_torque[3];   // rx,ry,rz torque
  };
  #pragma keylist CATForce

  struct CATPosition
  {
    double transformation[16]; // Last frame of manipulator
    double tcp_offset[16];     // Offset: last frame -> TCP
  };
  #pragma keylist CATPosition

  struct Location3D
  {
    double x;
    double y;
    double z;
  };

  struct BoxDimensions
  {
    double width;
    double height;
    double depth;
  };

  typedef double RotationMatrix[9];
```

```
  struct PointCloud
  {
    long                    id;        // pointCloud ID
    sequence<Location3D>    points;    // Point Positions
    boolean                 enabled;   // Pointcloud enabled?
    double                  spring;    // Spring constant
    double                  range;     // Path range
  };
  #pragma keylist PointCloud id

  struct CollisionBoundary
  {
    long                id;
    Location3D          position;      // XYZ-coordinates
    RotationMatrix      rotation;      // Rotation
    BoxDimensions       lengths;       // Box side lengths
    boolean             enabled;       // Enabled?
    double              spring;        // Spring stiffness
    double              damping;       // Damping
    boolean             end_effector;  // End effector?
  };
  #pragma keylist CollisionBoundary id
};
```

D. APPENDIX 4: DTP2 RHCS QOS SETTINGS

	QosPolicy	Field	CAT - EC / Default	CAT - VR	VR - OMS	EC - OMS	VR - EC
Topic	topic_data	value.length	0				
	durability	durability.kind	VOLATILE_ DURABILITY_ QOS	TRANSIENT_ DURABILITY_ QOS	TRANSIENT_ DURABILITY_ QOS	TRANSIENT_ DURABILITY_ QOS	TRANSIENT_ DURABILITY_ QOS
	durability_service	service_cleanup_ delay	0				
		history_kind	KEEP_LAST_ HISTORY_ QOS				
		history_depth	1				
		max_samples	LENGTH_UNLIMITED				
		max_instances	LENGTH_UNLIMITED				
		max_samples_ per_instance	LENGTH_UNLIMITED				
	deadline	period	DURATION_INFINITE				
	latency_budget	duration	0				
	liveliness	kind	AUTOMATIC_ LIVELINESS_QOS				
		lease_duration	DURATION_INFINITE				
	reliability	kind	BEST_EFFORT_ RELIABILITY_QOS	RELIABLE_ RELIABILITY_ QOS	RELIABLE_ RELIABILITY_ QOS	RELIABLE_ RELIABILITY_ QOS	
		max_blocking_ time	100 ms				
		synchronous	FALSE				
	destination_order	kind	BY_RECEPTION_ TIMESTAMP_ DESTINATIONORDER_ QOS				
	history	kind	KEEP_LAST_ HISTORY_QOS			KEEP_ALL_ HISTORY_QOS	
		depth	1				
	resource_limits	max_samples	LENGTH_UNLIMITED				
		max_instances	LENGTH_UNLIMITED				
		max_samples_ per_instance	LENGTH_UNLIMITED				
	transport_priority	value	0				
	lifespan	duration	DURATION_INFINITE				
	ownership	kind	SHARED_ OWNERSHIP_QOS				
Publisher	presentation	access_scope	INSTANCE_ PRESENTATION_QOS				
		coherent_access	FALSE				
		ordered_access	FALSE				
	partition	name.length	0				
	group_data	value.length	0				
	entity_factory	autoenable_ created_entities	TRUE				
Subscriber	presentation	access_scope	INSTANCE_ PRESENTATION_QOS				
		coherent_access	FALSE				
		ordered_access	FALSE				
	partition	name.length	0				
	group_data	value.length	0				
	entity_factory	autoenable_ created_entities	TRUE				

	QosPolicy	Field	CAT - EC / Default	CAT - VR	VR - OMS	EC - OMS	VR - EC
Datawriter	durability	kind	VOLATILE_ DURABILITY_QOS				
	deadline	period	DURATION_INFINITE				
	latency_budget	duration	0				
	liveliness	kind	AUTOMATIC_ LIVELINESS_QOS				
		lease_duration	DURATION_INFINITE				
	reliability	kind	BEST_EFFORT_ RELIABILITY_QOS				
		max_blocking_ time	100 ms				
		synchronous	FALSE				
	destination_order	kind	BY_RECEPTION_ TIMESTAMP_ DESTINATIONORDER_ QOS				
	history	kind	KEEP_LAST_ HISTORY_QOS				
		depth	1				
	resource_limits	max_samples	LENGTH_UNLIMITED				
	max_instances	max_instances	LENGTH_UNLIMITED				
		max_samples_ per_instance	LENGTH_UNLIMITED				
	transport_priority	value	0				
	lifespan	duration	DURATION_INFINITE				
	user_data	value.length	0				
	ownership	kind	SHARED_ OWNERSHIP_QOS				
	ownership_ strength	value	0				
	writer_data_ lifecycle	autodispose_ unregistered_ instances	TRUE				
Datareader	durability	kind	VOLATILE_ DURABILITY_QOS	TRANSIENT_ DURABILITY_ QOS			
	deadline	period	DURATION_INFINITE				
	latency_budget	duration	0				
	liveliness	kind	AUTOMATIC_ LIVELINESS_QOS				
		lease_duration	DURATION_INFINITE				
	reliability	kind	BEST_EFFORT_ RELIABILITY_QOS				
		max_blocking_ time	100 ms	RELIABLE_ RELIABILITY_ QOS			
		synchronous	FALSE				
	destination_order	kind	BY_RECEPTION_ TIMESTAMP_ DESTINATIONORDER_ QOS				
	history	kind	KEEP_LAST_ HISTORY_QOS				
		depth	1				
	resource_limits	max_samples	LENGTH_UNLIMITED				
		max_instances	LENGTH_UNLIMITED				
		max_samples_ per_instance	LENGTH_UNLIMITED				
	user_data	value.length	0				
	ownership	kind	SHARED_ OWNERSHIP_QOS				
	time_based_filter	minimum_separation	0				
	reader_data_ lifecycle	autopurge_nowriter_ samples_delay	DURATION_INFINITE				
		autopurge_disposed_ samples_delay	DURATION_INFINITE				

E. APPENDIX 5: TLX RATING SCALE DEFINITIONS SHEET

RATING SCALE DEFINITIONS		
Title	Endpoints	Descriptions
MENTAL DEMAND	*Low/High*	How much mental and perceptual activity was required (e.g., thinking, deciding, calculating, remembering, looking, searching, etc.)? Was the task easy or demanding, simple or complex, exacting or forgiving?
PHYSICAL DEMAND	*Low/High*	How much physical activity was required (e.g., pushing, pulling, turning, controlling, activating, etc.)? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?
TEMPORAL DEMAND	*Low/High*	How much time pressure did you feel due to the rate or pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic?
EFFORT	*Low/High*	How hard did you have to work (mentally and physically) to accomplish your level of performance?
PERFORMANCE	*Good/Poor*	How successful do you think you were in accomplishing the goals of the task set by the experimenter (or yourself)? How satisfied were you with your performance in accomplishing these goals?
FRUSTRATION LEVEL	*Low/High*	How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?

Figure E.1: NASA Task Load Index (TLX) Rating Scale Definitions [23].

F. APPENDIX 6: TLX WORKLOAD COMPARISON CARDS AND RATING SHEET

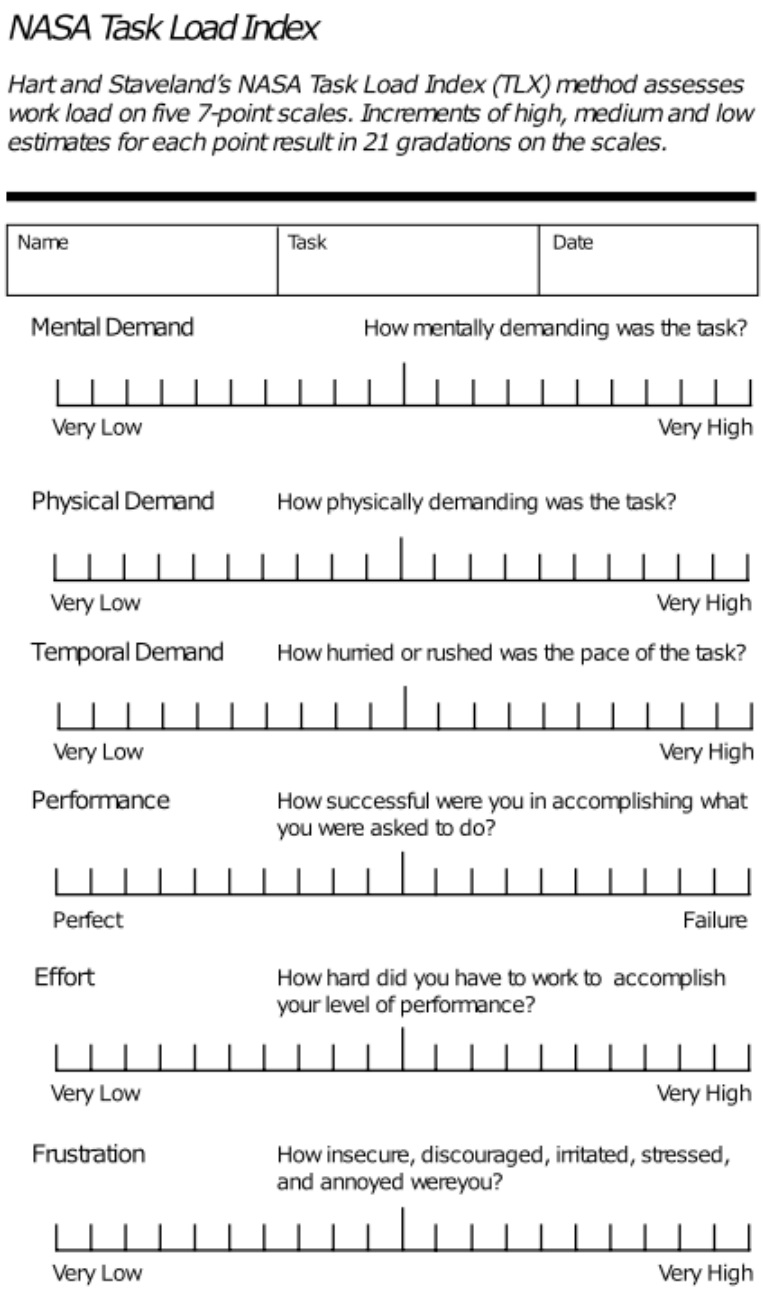


Figure F.1: NASA Task Load Index answer sheet [23].

Effort or Performance	Temporal Demand Or Frustration
Temporal Demand Or Effort	Physical Demand Or Frustration
Performance Or Frustration	Physical Demand Or Temporal Demand
Physical Demand Or Performance	Temporal Demand Or Mental Demand

Figure F.2: TLX workload comparison cards 1-8 [23].

Frustration Or Effort	Performance Or Mental Demand
Performance Or Temporal Demand	Mental Demand Or Effort
Mental Demand Or Physical Demand	Effort Or Physical Demand
Frustration Or Mental Demand	

Figure F.3: TLX workload comparison cards 9-15 [23].

G. APPENDIX 7: TLX SUBJECT INSTRUCTIONS

SUBJECT INSTRUCTIONS: RATING SCALES

We are not only interested in assessing your performance but also the experience you had during the different task conditions. Right now we are going to describe the technique that will be used to examine your experiences. In the most general sense we are examining the "workload" you experienced. Workload is difficult concept to define precisely, but a simple one to understand generally. The factors that influence your experience of workload mat come from the task itself, your feelings about your own performance, how much effort you put in, or the stress and frustration you felt. The workload contributed by different elements may change as you get more familiar with a task, perform easier or harder version of it, or move from one task to another. Physical components of workload are relatively easy to conceptualize and evaluate. However, the mental components of workload may be more difficult to measure.

Since workload is something that is experienced individually by each person. There are no effective "rulers" that can be used to estimate the workload of different activities. One way to find out about workload is to ask people to describe the feelings they experienced. Because workload may be caused by many different factors, we would like you to evaluate several of them individually rather than lumping them into a single global evaluation of overall workload. This set of six rating scales was developed for you to use in evaluating your experiences during different tasks. Please read the descriptions of the scales carefully. If you have a question about any of the scales in the table, please ask me about it. It is extremely important that they be clear to you. You may keep the descriptions with you for reference during the experiment.

After performing each of the tasks, you will be given a sheet of rating scales. You will evaluate the task by putting an "X" on each of the six scales at the point which matches your experience. Each line has two endpoint descriptors that describe the scale. Note that "own performance" goes from "good" on the left to "bad" on the right. This order has been confusing for some people. Please consider your responses carefully in distinguishing among the different task conditions. Consider each scale individually. Your ratings will play an important role in the evaluation being conducted, thus, your active participation is essential to the success of this experiment and is greatly appreciated by all of us.

Figure G.1: TLX subject instructions sheet for rating scales [23].

SUBJECT INSTRUCTIONS: SOURCES-OF-WORKLOAD EVALUATION

Throughout this experiment the rating scales will be used to assess your experiences in different task conditions. Scales of this sort are extremely useful, but their utility suffers from the tendency people have to interpret them in individual ways. For example, some people feel that mental or temporal demands are the essential aspects of workload regardless of the effort they expended or the performance they achieved. Others feel that if they performed well the workload must be low and vice versa. Yet others feel that effort or feelings of frustration are the most important factors in workload and so on. The results of previous studies have already found every conceivable pattern of values. In addition, the factors that create levels of workload differ depending on the task. For example, some tasks might be difficult because they must be completed very quickly. Others may seem easy or hard because of the intensity of mental or physical effort required. Yet others feel difficult because they cannot be performed well, no matter how much effort is expended.

The evaluation you are about to perform is a technique that has been developed by NASA to assess the relative importance of six factors in determining how much workload you experienced. The procedure is simple. You will be presented with a series of pairs of rating scale titles (for example Effort vs. Mental Demands) and asked to choose which of the items was more important to **your** experience of workload in the task(s) that you just performed. Each pair of scale titles will appear on separate card.

> **Circle the Scale Title that represents the more important contributor to workload for the specific task(s) you performed in this experiment.**

After you have finished the entire series we will be able to use the pattern of your choices to create a weighted combination of the ratings from that task into a summary workload score. Please consider your choices carefully and make them consistent with how you used the rating scales during the particular task you were asked to evaluate. Don't think that there is any correct pattern: We are only interested in your opinions.

If you have any questions, please ask them now. Otherwise, start whenever you are ready. Thank you for your participation.

Figure G.2: TLX subject instructions sheet for sources-of-workload evaluation cards [23].